SPECTRUM

VILÁG IV. rész



1. JÁTÉKTÖRTÉNELEM

Sok víz lefolyt már azóta a Dunán, amióta az első játékprogramok megjelentek a piacon Spectrum számítógépre. Az azóta eltelt mintegy öt év alatt a programozástechnika sokszoros fejlődésével öszehasonlíthatatlanul jobb program-"alkotások" születtek az előzőeknél, s az útnak még messze nincs vége.
Tekintsünk vissza a kezdet kezdetére. A Spectrumosoknak minden új szenzációnak számított, így hazai viszonylatban is örülhettünk, ha egy-egy új programhoz hozzá tudtunk jutni. Az első kazettáinkon mindenféle "saláta" megtalálható volt, begépeltük az újságok hasábjain található utolsó BASIC programot is, mindent felvettünk függetlenül attól, hogy tudtuk-e használni, vagy sem. Az első "nagy" 16K-s játékok (pl. PSION-Space Raiders, vagy ULTIMATE-Jet Pac) után egyre több 48K-s játék látott napvilágot. A kezdeti időszakban egy-egy játék-forgalmazó szinte monopóliumot jelentett a piacon (pl. PSION, ULTIMATE, ARTIC, Dk'TRONICS, IMAGINE), de egyre több ismeretlen név bukkant fel, és hívta fel magára a figyelmet kimagasló tevékenységével. 1984-re már kezdett telített lenni a piac, újabb és újabb játékötletek láttak napvilágot. Egyre több szimulációs és sportjáték, kalandjáték vált híressé, megindult a harc az egyes software-házak között, hogy azonos témában ki alkot jobbat. Ha egy témát már elsütöttek, nagy bátorság, és összehasonlíthatatlanul magasabb színvonalú kivitelezés kellett ahhoz, hogy az adott ötlet újra forgalomba kerüljön és elsöpörje az előzőt. Ilyen harc alakult ki a sportjátékok terén is. Hazai felhasználók körében is jelentős változást tapasztalhattunk, kezdtük megválogatni, mit veszünk fel, és mit dobunk ki. Bár természetesen akadtak "harácsolók", akik a legutolsó "szemetet" is begyűjtötték, és lehet, hogy igazuk volt, hiszen a sok "egyéb" között lapulhatott igazi érték is. Ez csak később derült ki: 1984 és 1987 között mérföldes léptekkel haladt előre a programok minőségének javulása, a 16K-s programok elmaradtak, a 48K-s programok között 'szupercsodák' jelentek meg. Igen népszerűvé váltak az úgynevezett ULTIMATE klónok, vagyis a 3D "szobás" játékok, melyeknek újabb és újabb tagjai folyamatosan látnak napvilágot. A kalandjátékok láza egy kicsit visszaesett, bár ma is



készülnek bonyolult konstrukciók ebből a játéktípusból, de ebben a kategóriában sok újat már nem tudnak kitalálni. A szimulációs játékok özöne is véget ér talán, egyedül az akciójátékok körében érezhető újra fellobbanás, és úgy tűnik, az érdeklődést egy ideig még fenn tudják tartani.
A játékok fejlődése során szebbnél-szebb screen-ekkel találkozhattunk; az utóbbi időben csak ültünk és füleltünk, hogy egyetlen szerencsétlen BEEP-csatornával milyen szenzációs többszólamú zenét képesek alkotni (pl. PING-PONG); sőt 1987 a 128K-s Spectrum-tulajdonosok számára is hozott újdonságot, ugyanis több 48K-s játék a 128K-s gépen 3 szólamban zenél (pl. ELEVATOR ACTION, THE TUBE, GLIDER RIDER).
A játékok fejlődésével egyre trükkösebb védelmi formákkal találkozhatunk. A kezdet-kezdetén nem tűnt fel jelentősebb védelem. Később elsősorban az 'OCEAN' jóvoltából megjelent a Speed Lock turbósítás, majd egyre több turbós, Jerky-s (szaggatott TONE), rövid TONE-os (pl. FAIRLIGHT) software látott napvilágot. Ma már ezek a trükkök mind feltörhetők a 'MULTIFACE ONE' hardware eszközzel, régebben csak a TC-6, TC-7, esetleg elmélyült gépi kódú fejtegetés segíthetett. Sok gyári program védelem nélkül indult el, de útközben "megpiszkálták" - erre főleg Jugoszláviában került sor - de, mint ahogy minden szernek van ellenszere, ezeket a feladatokat is hamar megoldották honfitársaink.
Mindezeket összegezve megállapíthatjuk, hogy a cérna végén másolható program jut el a felhasználóhoz, még akkor is, ha menet közben kötöttek egy-két csomót a cérnára.

2. SOFTWARE BANK

2.1 PROGRAMISMERTETŐ

The Golden Mask - COMPASS - Kaland

A "MORRACK" kalandtörténet feldolgozását vállalta magára a Compass cég, amikor januárban megjelentette a történet első részét 'Demon From the Darkside' címmel. Akik ismerik ezt a kalandot és végig is játszották, azok tudják, hogy MORRACK-nak, ASHMEARD varázsló tanítványának feladata 'LORD DRAKON a gonosz' megsemisítése. Ez látszólag sikerül is, de a játék végén állandó kísérőnk, a bagoly nyomatékosan felhívja a figyelmünket, hogy DRAKON még él és mozog. A 'GOLDEN MASK' a történet folytatása. Meg kell szereznünk az arany álarcot, mert csak így van hatalmunk a szívós DRAKONT legyőzni, mielőtt véghezvinné csúnya gonosz terveit országunkban, a DRAL-ok földjén. Érdekes történet, izgalmas kaland.

Howard the Duck - ACTIVISION - Akció

A CARTOON gondozásában megjelent Marvel Comics történetek egyik kedvence HOWARD kacsa. Daffy-tól és Donald kacsától eltérően HOWARD-ban az a különös, hogy nem éppen "jófiú". Szereti a nőket, iszik, napi 20 szál cigarettát elszív, egyszóval nagyvilági életet él. A HOWARD sorozat 'Adventure of Volcano Island' című epizódját dolgozza fel a játék, amely Vulkán szigetén játszódik. A szigetet egy sebes sodrású, rendkívül kanyargós folyó kettészelte, a hegy a túloldalon van, így a kacsa első feladata a folyón átkelni. Ha feljut a hegyre, át kell mennie egy hídon, amely a vulkánhoz vezet. Itt a közlekedést a magasból potyogó sziklák kissé nehezítik, gyakorló fokozat esetén a játék itt véget is ér. Ha a profi fokozatot választjuk, még el kell repülni egy repülőgéppel a vulkán felett a sötét 'Overlord' kapujáig. A további rejtelmek feltárását átengedjük Olvasóinknak.
Élvezetes játék, nem túl kiemelkedő grafikával.

Catch 23 - MARTECH - Akció

Kiváló 3D vonalas grafikával megalkotott akciójáték. Játékosunknak be kell 'szivárognia' a CK23 fedőnevű kísérleti kutatótelepre, amely a bolygó legtitkosabb hadikomplexuma. Miután lehajítják a főhőst a szigetre, az akadályok özönét kell leküzdenie. Át kell másznia magas hegyeken, túljutnia halált hozó aknákon, nagyfeszültségű elektromos kerítéseken, el kell bújnia az itt-ott elrejtett figyelő szemű kamerák elől. Megfeszített figyelemmel kell hősünket irányítanunk, mert bármikor váratlan meglepetések érhetik. A bázisra érkezve az épületeken belül újabb nehézségekkel kell szembenéznie. Meg kell keresnie a nukleáris reaktort, és itt el kell helyeznie egy időzített bombát, majd a legnagyobb lehetséges sebességgel távoznia kell a tett színhelyéről és a szigetről.
Izgalmas, hosszú játék, azoknak ajánljuk, akik rá tudják szánni az időt, érdemes!

The Last Ninja - SYSTEM 3 - Akció

A Ninja történetek egész sorát játszottuk végig az utóbbi hónapokban, de a sorozat még nem ért véget. A System 3 újabb mesével rukkolt ki. A Ninja harcost 125 képernyőn - pusztaságon, gyönyörű kerteken, félelmetes labirintuson - keresztül vezetve kell eljuttatnunk a célhoz. Útközben a dobócsillagon, kardon, lándzsán, BO-n kívül csak a szerencsénkben bízhatunk, mivel elmaradhatatlan ellenségeink (karddal hadonászó szamuráj, karatebajnok, kapuőrség, stb.) váratlanul fel-felbukkannak, és nézeteltéréseinket tisztázni kell velük, utunk folytatása előtt.
Végcélunk, hogy megtaláljuk és legyőzzük a SOGUN-t.
A játék stílusa hagyományos, a háttérgrafika kiváló.

International Events-ANCO-Szimuláció

Bizonyára már mindenkinek a könyokén jönnek ki a mindenféle sportjáték és egyéb szimulációk. Az Anco ennek ellenére pozitívan bizonyított. Hat különböző sportág feldolgozásával hívták fel magukra a figyelmet. A sárkányrepülés, wind-surf, motorverseny, cross, vizisí, és a motoros ügyességi ugrás ilyen együttes összeállítása még nem szerepelt. A versenyeket játszhatjuk egyedül, a géppel, vagy egymás között is. A grafikai kivitelezés jó színvonalú, de a játék sok új ötletet nem tartalmaz, így esetleg egy fárasztó programozás közben felüdülést nyújthat.

Beta Basic 4.0-BETASOFT-Basic bővítő

A Betasoft programozói gárdája elsősorban a 128K géptulajdonosoknak próbált kedvezni, amikor elkészítette a Beta Basic legújabb verzióját. A Beta BASIC 4.0 mindent tud, amit elődje, a 3.0, valamint a következő jó tulajdonságokkal rendelkezik:

- 64K-nál nagyobb tömb dimenzionálható (pl. DIM! tomb (600,100)); bármelyik eleme elérhető 0,2 mp alatt az INARRAY funkcióval; ugyanennek a tömbnek a sorbarendezése mindössze 5 mp-et vesz igénybe
- a LIST! és INPUT! új utasítások segítségével közvetlenül manipulálhatunk a RAM disc és a Microdrive között az aktuális memória igénybevétele nélkül
- a 3 csatornás hang megszakításvezérelt meghajtását BASIC utasítások teszik lehetővé
- a FILL rutin gyorsabb lett, ugyancsak gyorsabb lett két rajzoló utasítás is (a kör rajzolás 12-szer, a DRAW üzemmód 2,5-szer)
- BASIC-ből szelektálható az RS-232/48K normál printer output
- a SAVE! segítségével a BASIC program bármely meghatározott részét a RAM disc-re menthetjük.

Reméljük, hogy az új verzió megnyeri a felhasználók tetszését, bár igazán csak a 128K tulajdonosok jártak jól vele.

2.2 JÁTÉK POKE



Krakout-Gremlin Graphics

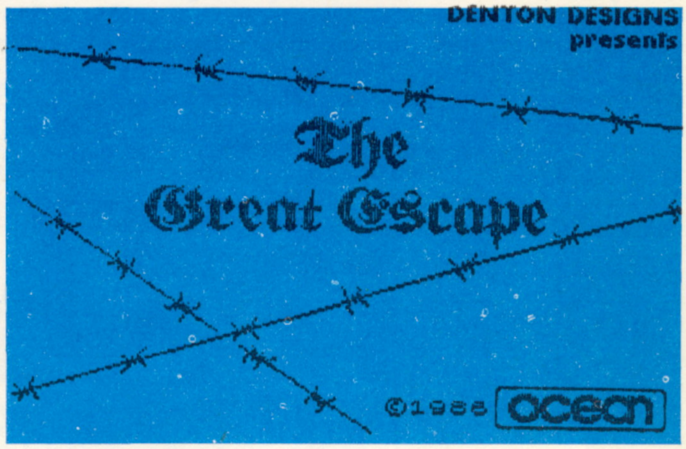
Az 'ARKANOID'-klónok sorába tartozó programmal valóban kedvezőbb eredményeket érhetünk el végtelen golyóval. Ehhez a 46565. címen kell zérust elhelyeznünk.
A bevitel csak azok számára lesz hasznos, akik a kazettán a következő file-térképpel rendelkeznek: 164/6913/20000/20536/6916/1704.
A BASIC betöltő után állítsuk le a magnetofont, majd BREAK-eljük le a programot. Ezután állítsuk át a tinta-színt: INK 7 és nyomjuk meg kétszer az ENTER-t.
Adjuk ki: CLEAR (ENTER), majd gépeljük be 20 (ENTER), generálva ezzel egy üres sort.
Állítsuk át a kezdősor hosszát: POKE 23757,96 (ENTER), erre a 20.sor eltűnik, de helyére beírhatjuk a szükséges byte-okat, ehhez pedig gépeljük be a következő sorokat:

```
20 FOR i=23841 TO 23853: READ a: POKE i,a: NEXT i
30 DATA 62,0,50,229,181,62,255,55,229,195,86,5,0
```

Futtassuk a programot a 20. sortól: RUN 20 (ENTER), majd a 20. és 30. sort törölhetjük. A végtelen életet generáló kód már a betöltőben van.
Adjuk ki: CLEAR 24600 (ENTER), majd futtassuk ismét a betöltőt:
RANDOMIZE USR 23760 (ENTER), és indítsuk újra a magnetofont. Betöltődés után annyi golyónk lesz, 'mint égen a csillag'.

2.3 THE GREAT ESCAPE

Az OCEAN 1986 közepén jelentkezett a piacon a GREAT ESCAPE című film hasoncímű számítógépes átiratával. A program három dimenziós grafikai kidolgozása sok hasonlóságot mutat a korábban megjelent IMAGINE-programmal, a MOVIE-val. A játékos mozgatását a programozók rendkívül érdekesen oldották meg, mert nem a figura mozog a pályán, hanem a PÁLYA MOZOG A FIGURA ALATT!



A történet 1942-ben játszódik a colditz-i német hadifogolytáborban. A tábort az afrikai fronton fogságba esett angol katonák és a Németország felett lelőtt bombázógépek személyzete népesíti be. A játékban is egy ilyen fogoly angol pilótát személyesítünk meg, feladatunk a táborban található eszközök segítségével sikeresen megszökni.

Sikeres betöltés után megjelenik a főmenü, ahol billentyűzet- és joystick-irányítás között választhatunk, illetve definiálhatjuk az irányító billentyűket. A játék kezdetén édesdeden szundikálunk egy osztályon felüli szálloda (német hadifogolytábor) baldachinos ágyikóján. Most automatikus fogoly üzemmódban vagyunk: a játékost szimbolizáló figura (mindig a képernyő közepén látható) a társaival együtt végzi a hadifoglyok napi teendőit (alvás, sorakozó, étkezés stb.). Ez a viselkedésforma kezdőknél kiválóan alkalmas a tábor megismerésére (a mellékelt térkép használatával ez jelentősen meggyorsul).

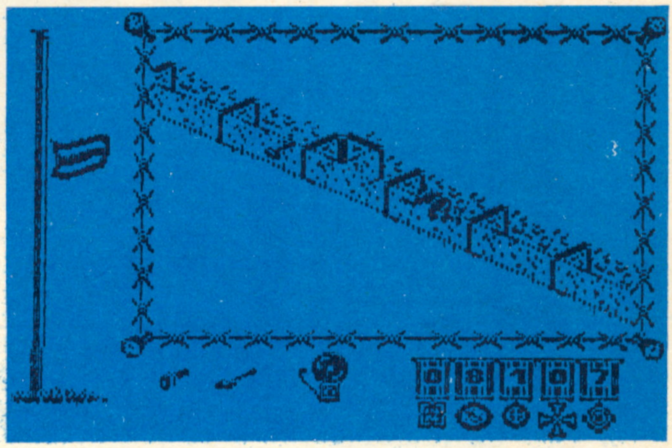
Ha a sok semmittevés után már elég lelki indíttatást érzünk magunkban, bármely irányító billentyű megnyomásával kitörhetünk a fogoly üzemmódból és megkezdhetjük a szökéshez szükséges tárgyak összegyűjtését. A tárgyak a tábor területén szétszórva, a térkép által jelzett helyeken találhatóak. Felvételük a 'tűz' és a 'fel', eldobásuk a 'tűz' és a 'le' gombok együttes megnyomásával lehetséges. Egyszerre maximum két tárgy lehet nálunk, amelyeket a képernyő alsó részén, a csengőtől balra láthatunk. A bal oldali tárgy aktiválása a 'tűz+balra', a jobb oldalié a 'tűz+jobbra' billentyűk megnyomásával lehetséges. Ha valamely tárgyra már nincs szükségünk, rejtsük el valami olyan helyen, ahol nem találhatják meg az őrök. A tábor rendszabálya alapján a fogolynál nem lehet semmilyen tárgy, ha az őrök ilyet találnak nálunk kíméletlenül bekóteroznak és a nálunk lévő tárgy visszakerül arra a helyre, ahonnan felvettük. Ha az őrök csak az elrejtett tárgyat találják meg, az visszakerül az eredeti helyére.

Ha önálló kalandokra vállalkozunk, feltétlenül kísérjük figyelemmel a csengő jelzéseit, amelynek rövid idejű csörgése - felirattal együtt (ld. a szótárt a leírás végén) - valamely elvégzendő "fogoly-teendő"-re hívja fel a figyelmünket, de a folyamatos csörgés már a külön utakon történő mászkálásunkból kifolyólag elrendelt riadót jelzi. Riadó esetén azonnal csatlakozzunk a többi napi teendőjét végző fogolyhoz, mert rövidesen a kóterben találhatjuk magunkat.

A kóter (nem katonaviseltek kedvéért: kóter = magánzárka, cella) - úgy tűnik - külön a számunkra létesített intézmény: állandóan itt találjuk magunkat, mihelyt elmulasz-

tunk valamit fogolykötelességeink közül vagy olyan időben (pl. éjszaka) mászkálunk valami olyan helyen, ahol éppen semmi keresnivalónk sincsen (pl. a szögesdrót kerítések között). A cellában kell töltenünk az egész éjszakát és csak másnap reggel, a táborparancsnok kíséretében engednek ki minket a sorakozóra.

A bekóterozásnak kellemetlen következményei vannak: a képernyő bal oldalán látható - eleinte felhúzott - zászló lejjebb ereszkedik. A zászló jelzi a hangulatunkat, vagyis azt, hogy mennyi esélyünk van a kitörésre az automatikus fogoly üzemmódból (mennyire engedelmeskedik a figuránk az irányító billentyűknek) illetve piros színnel figyelmeztet arra, hogy tiltott zónában járunk. A hangulat néha a bányabéka ülepe alá esik, ekkor azonnal tenni kell valamit a szabadulás érdekében, mert egy sikeres cselekmény serkentőleg hat lelkivilágunkra (másrészt emeli a képernyő jobb alsó sarkában, a kitüntetések szalagján látható pontszámunkat), feljebb megy a zászló. Kiváló módszer például az álruhát gyors egymásutánban levetni-felvenni. Ezt azért lehetőleg ne német őr előtt végezzük, mert hevenyészett divatbemutatónkat némi bekóterozással fogja díjazni.



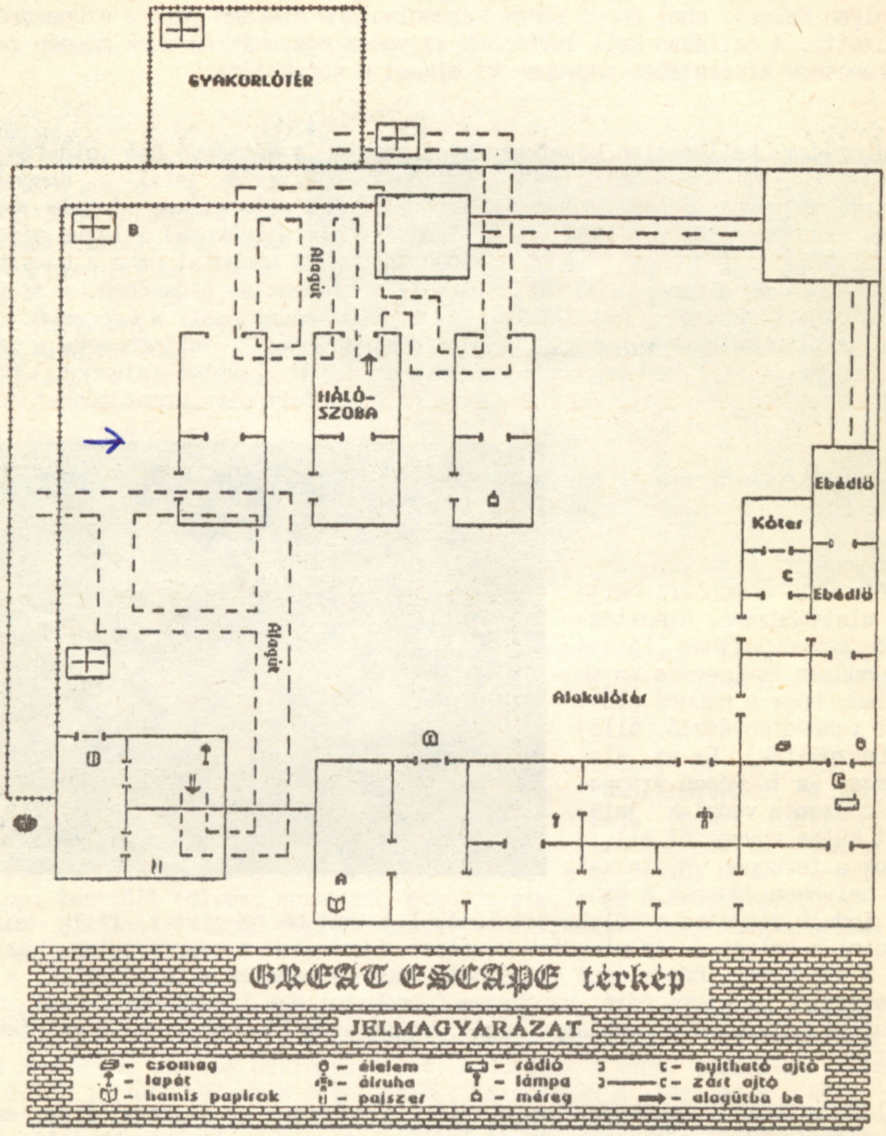
A német hadifogolytáborok - ha történelmi ismereteink nem tévesek - híresek voltak a foglyoknak nyújtott maximális kényelemről. Ezt demonstrálhatja a tábor alatt húzódó, kiépítésre váró metróvonalnak látszó alagútrendszer is (persze az se lehetetlen, hogy a háború előtt a tábort vakondtenyésztő állomásnak használták). Ez az alagútrendszer (a térképen szaggatott, párhuzamos vonalak jelölik) két külön egységből áll, a bejáratok a térképen nyilakkal jelzett helyeken vannnak. A szobánkból induló alagútba a kályha eltolásával juthatunk be (amit illik kultúráltan visszatolni a helyére). Az alagutak kiválóan alkalmasak a veszélytelen közlekedésre (később a szökésre), bár eléggé elítélhető módon a németek elfelejtették a világítást beszerelni és kissé sötét van benne (mondhatni nem látunk egy kukkot se). Eme tarthatatlan helyzeten a zseblámpa segít, világosság lőn (mint azt a mellékelt ábra is mutatja).

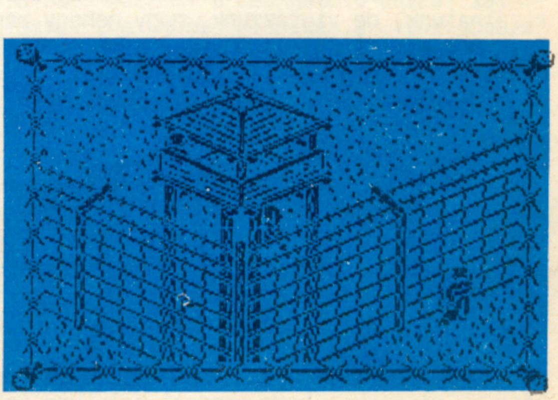
Szabadulást elősegítő tevékenységre kiválóan alkalmasak az éjszakai órák, amikor jótékony sötétség borul a táborra. Az éj leple alatt viszonylag kisebb feltűnéssel kóricálhatunk, de vigyázzunk, hogy nehogy ránk essen az őrtoronyból világító fénypászma, mert riadót fújnak. Ha folyamatosan jelez a csengő, pucoljunk vissza az ágyikóhoz és ott engedjük el a billentyűket, hogy visszaálljunk automatikus fogoly üzemmódba (lefekszünk). Így a razzia már szende szundikálásba merülve talál bennünket (különben irány a kóter).

A GREAT ESCAPE nem olyan játék, mint például a PYJAMARAMA, ahol minden újabb játéknál ugyanazokat a fázisokat kell végigjátszanunk. Itt - akárcsak a stratégiai játékoknál - majdnem teljesen a játékostól függ, hogy milyen tárgyakat használ fel, milyen útvonalat választ stb. A program eme sajátossága miatt nem is adunk "lépésről lépésre receptet" a végcél eléréséhez (a stratégia kidolgozása legyen a játékos dolga), de ismertetjük az összes tárgy funkcióját.

Mindenekelőtt még egyszer megjegyeznénk, hogy bekóterozásunk esetén a nálunk lévő tárgyak visszakerülnek eredeti helyükre (elrejtett tárgy felfedezése esetén csak az, amelyet megtalálták az őrök) és az általunk kinyitott ajtókat újra bezárják (kulcs az eredeti helyén).

2.3 THE GREAT ESCAPE

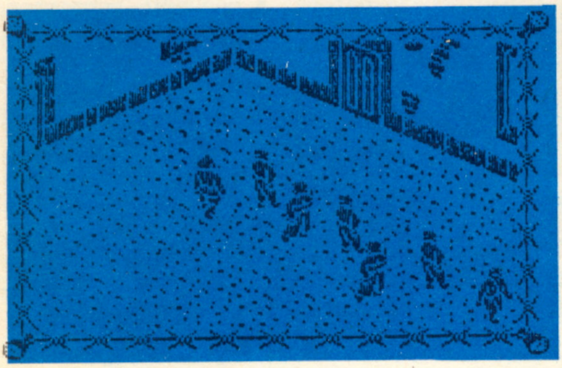




Kulcsok: Van néhány szoba, amelynek ajtaját zárva tartják (a térkép jelöli ezeket), de kulccsal bejuthatunk. A fehér betűkkel jelzett zárt ajtókat a fekete betűvel jelölt helyen lévő kulcsokkal kinyithatjuk, az A ajtót az A kulccsal és így tovább. A képen éppen az őrtorony lábánál levő B kulcs megszerzésén mesterkedünk, az őr vizsla tekintetétől kísérve.

Lapát: Az alagútrendszer elég kezdetleges technológiával épülhetett, mert néha beomlik. Ilyen helyekeh a közlekedés némi akadályokba ütközik, de a lapát használatával az omlás eltávolítható. Megjegyeznénk, hogy bekóterozásaink ünnepi alkalmaival a már eltakarított torlaszok visszakerülnek helyükre.

Pajszer: ha nem kívánunk mindenféle kulcsokkal bíbelődni, kissé drasztikus módon, a pajszer nevű eszközzel nyithatjuk ki a bezárt ajtókat, bár ez némileg hosszabb időt vesz igénybe, mint a kulcsos szisztéma. A pajszer mindenképpen hasznos szerszám pl. az A ajtónál, ahol a nyitásra szolgáló kulcs az ajtó meletti szobában van (a hamis papírokkal egyetemben), bár ez a szoba a fél világ körbejárása után a C ajtón keresztül is elérhető. Érdekes, hogy lehetőség van az egyébként nyitott ajtók feszegetésére is (a program megteszi), bár ez a cselekvés már kimeríti a heveny elmebaj fogalmát...

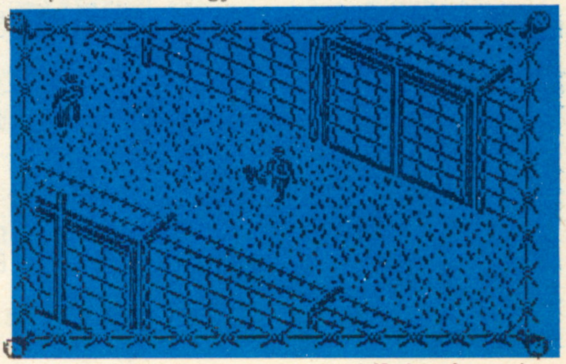


Álruha: Német egyenruhában nyugodtan mászkálhatunk a táborban, az őrök kollégának néznek bennünket, kivéve a táborparancsnokot, akit tányérsapkájáról ismerhetünk fel (a képen látható sorakozónál Ő a bal szélen látható kedves kis figura). A parancsnok egy szőrös szívű huligán, aki - véletlenül - mindig arra jár, ahol éppen valami tiltott dolgot művelünk (pl. éppen kijövünk az alagútból). Teljesen érthetetlen módon ennek az unszimpatikus úriembernek az a hobbija, hogy mindig kóterbe csuk bennünket és - úgy tűnik - erről a megrögzött szokásáról az istennek se hajlandó lemondani.
Megszívlelendő jótanács: szökéskor ne legyen rajtunk német egyenruha, mert kivégeznek minket mint kémet!

Lámpa: Az alagút (nem a budapesti!) kivilágítására alkalmas Holdvilág V0.0 márkájú kiváló szerkezet. Tapasztaltabbak (akik már ismerik az alagút kanyarjait) és macskák számára nem szükséges.

Rádió: Ez valóban az egyik leghasznosabb kütyü a játékban. Aki esetleg totózik, vasárnaponként meghallgathatja a totóeredményeket és hogy mit játszott a Fradi szombaton. Dózsa-drukkereknek nem dukál rádió.

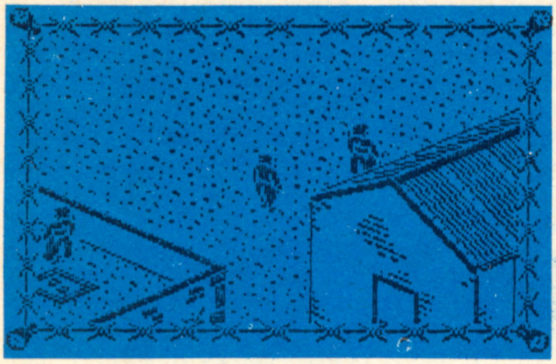
Élelem: Nem sikerült pontosan kideríteni, hogy milyen célokat szolgálhat (mert minden nap van étkezés), de mindenesetre azt jótékony homály fedi, hogy a - láthatóan - konzervált élelemhez konzervnyitót vajon honnan lehet szerezni (mint az köztudomású enélkül az elmés szerkezet nélkül a konzerv kissé nehézkesen fogyasztható). Erős gyomorsavval rendelkező foglyok esetében ez a probléma tárgytalan...



Méreg: Használatával az élelem megmérgezhető és finom angol specialitásként átnyújtható az őr uraknak. Ha elfogadják, kiváló jelenetnek lehetünk tanúi: az őr úgy látszik nem rajong a modern gasztronómia legújabb vívmányaiért, mert heveny haláltusa után kiszenved. Tudomásunk szerint az Állatvédő Liga nem tart fennt kirendeltséget a colditzi fogolytáborban, szóval ezt a tréfát eljátszhatjuk a szögesdrót kerítések között cirkuláló kedves vérebekkel is (pont ilyen mesterkedés folyik a képen is). Mérgezett étellel vesztegetni fogolytársainkat nem túl humánus dolog, bár ez valószínűleg senkinek sem fordult meg a fejében.

Csomag: A Nemzetközi Vöröskereszt szeretetcsomagjaiban érkeznek a szökéshez szükséges legfontosabb tárgyak. Az elsőben egy zacskó cukorka (na ez az, ami nem annyira fontos), a másodikban egy nélkülözhetetlen eszköz: a csípőfogó a drótkerítés átvágá-

2.3 THE GREAT ESCAPE



sához (a főbejáraton sajnos nem engednek ki minket, kénytelenek vagyunk dróton át szökni). A harmadik csomagban egy csokoládészelet érkezik, amellyel megvesztegethetünk egy másik foglyot. A lekenyerezésből (lecsokoládézásból) kifolyólag a másik rab destruktív magatartásával magára vonja az őrök figyelmét, mialatt mi vígan megszökhetünk. A negyedik csomagban iránytű érkezik, a szökésnél szintén nagyon fontos.

A csomagok folyamatosan, minden reggel érkeznek (mindig ugyanabban a sorrendben) és a kibontás után gondosan el kell rejtenünk a benne lévő tárgyat (lehetőleg az alagútba), mert ha felfedezik vagy elkobozzák tőlünk, a következő csomagban ugyanazt a tárgyat látjuk viszont (a kóterozás eredményeképpen már kaptam egyszer 22 zacskó cukrot, de az a kellemetlen táborparancsnok nem engedélyezte, hogy trafikot nyissak a hadifogolytábor közepén). A csípőfogóval különösen vigyázzunk, mert az őrök ezt még automatikus fogoly üzemmódban is felfedezik nálunk és mehetünk a törzshelyünkre (a kóterbe).

Tippek a szökéshez (TIPP-TOPP PARÁDÉ):

Kézenfekvőbb a hálószobánkból kiinduló alagút használata, vagyis a gyakorlótéren keresztül szökni, mert ott csak egy őr van. Természetesen lehetőleg ne gyakorlatozási időben szökjünk, mert akkor némileg nagyobb az őrpopuláció és elég feltűnő lesz, ha egyetlen nekirugaszkodással szanaszét vágjuk a kerítést. A másik alagút a szögesdrótok közé vezet. Ott is meg lehet szökni, de sok az őr meg a kutya, vagyis nagyobb a rizikó.

Feltétlenül gyűjtsük össze a kiválasztott alagútba a következő tárgyakat: lámpa, mert nem árt ha látunk is valamit; lapát az esetleges torlaszok eltávolítására; csípőfogó a szögesdrót átvágásához; iránytű, hogy sikeres szökés után tudjunk tájékozódni; hamis papírok a határ eléréséig következő igazoltatások kicselezésére.

Várjuk meg, a gyakorlatozási idő végét (esetleg megvesztegethetjük az egyik társunkat a csokoládéval, hogy játsszon csalétket), menjünk ki az alagúton keresztül a gyakorlótérre, zsebünkben a csípőfogóval és az iránytűvel. Vágjuk át a drótot, tegyük le az iránytűt, majd menjünk vissza a hamis papírokért. Újból kimenve a kerítésen dobjuk el a csípőfogót, vegyük fel az iránytűt és szaladjunk a képernyő széle felé, ahol már vár minket a SZABADSÁG...

Ha megfelelően felszerelve (a fenti cuccokkal) szöktünk meg a program közli velünk, hogy

WELL DONE
YOU HAVE ESCAPED FROM THE CAMP
AND WILL CROSS THE BORDER SUCCESSFULLY

vagyis hogy jól van, sikerült megszöknünk a táborból és szerencsésen el fogjuk érni a határt.

Ha nem megfelelő felszereléssel szökünk meg, akkor megtudhatjuk, hogy ügyesek vagyunk, megszöktünk, de az első őrjárat visszahozott bennünket, mert nem voltunk kellően felszerelve (lehet tovább próbálkozni). Ha mindezt német egyenruhába öltözve adjuk elő, akkor már vissza se hoznak, hanem szerveznek nekünk egy exkluzív találkát a kaszással (tőmondatban: kivégeznek minket, mint kémet).

Végezetül közölnénk a képernyő bal alsó részén megjelenő üzenetek jelentését:

YOU ARE IN SOLIDITARY	- A kóterben vagyunk
WAIT FOR RELEASE	- Szabadulásra várunk
ANOTHER DAY DAWNS	- Új nap kezdődött
TIME TO WAKE UP	- Ébresztő
RED CROSS PARCEL	- Vöröskereszt-csomag érkezett
YOU OPEN THE BOX	- Kinyitottuk a csomagot
ROLL CALL	- Sorakozó, irány az alakulótér
MISSED ROLL CALL	- Nem jelentünk meg a sorakozónál
BREAKFAST TIME	- Reggeli, irány az ebédlő
THE DOOR IS LOCKED	- Az ajtó zárva van
INCORRECT KEY	- Rossz kulccsal próbálkozunk
PICKING THE LOCK	- A pajszerral feszítjük az ajtót
THE DOOR IS OPENED	- Kinyílt az ajtó
ITEM DISCOVERED	- Felfedezték az elrejtett tárgyat
EXERCISE TIME	- Gyakorlatozási idő
HE TAKES THE BRIBE AND ACTS AS DECOY	- A rabtársunk elfogadta a csokoládét és hajlandó lesz magára vonni az őrség figyelmét, amíg megszökünk

Kellemes szökdösést!

Folytatás a 25. oldalról:

```
 10 LET a=50000
 20 FOR i=40000 TO 40255
 30 GO SUB 100
 32 POKE a+10,PEEK (i+6144)
 35 LET a=a+11
 40 NEXT i
 50 FOR i=42048 TO 42303
 60 GO SUB 100
 62 POKE a+10,PEEK (i+4352)
 65 LET a=a+11
 70 NEXT i
 80 FOR i=44096 TO 44351
 90 GO SUB 100
 92 POKE a+10,PEEK (i+2560)
 95 LET a=a+11
 96 NEXT i
 98 STOP
100 RANDOMIZE i-23616
110 POKE a,PEEK 23670: POKE a+1,PEEK
    23671
120 POKE a+2,PEEK i: POKE a+3,PEEK (i+
    256)
130 POKE a+4,PEEK (i+512): POKE a+5,
    PEEK (i+768)
140 POKE a+6,PEEK (i+1024): POKE a+7,
    PEEK (i+1280)
150 POKE a+8,PEEK (i+1536): POKE a+9,
    PEEK (i+1792)
160 RETURN
```

A BASIC program segítségével a 40000. címtől 6912 byte hosszú képernyő file a visszatöltéshez megfelelő állapotban menthető ki az 50000. címtől kezdve, hossza megnőtt, 8448 byte lett.
Most mentsük ki az új képernyőnket:

```
SAVE "ujkep" CODE 50000,8448
```

úgy, hogy a magnó pillanat-stop gombját csak a fejléc kitöltése után oldjuk ki, mert a fejlécre nem lesz szükségünk.
Itt szöget üthet a fejünkbe, hogy vajon, ha fejléc nincs, és a programban sem adtuk meg, honnan fogja tudni a rutin, hogy mikor ne töltsön tovább, mikor van vége a betöltésnek?
Nos az átviteli jelzőbit a betöltés ideje alatt mindvégig 1 értékű lesz, a betöltés végén viszont zérus, ez pedig egy 'RET NC' következtében visszatérést jelent a szubrutinból.
Ha elhelyeztük a rutint a memóriában, adjuk ki:

```
RANDOMIZE USR 40000
```

és töltsük be az előzőleg kimentett - 8448 byte hosszú - fejléc nélküli kódot.
Íme: az eredmény jól látható!

2.4 SILENT SERVICE

A Microprose/US GOLD cég 1986-ban kiadott SILENT SERVICE című programja a HUNTER KILLER óta megjelent legjobb tengeralattjáró-szimuláció. Grafikai kidolgozása és logikája követi a US GOLD által forgalmazott szimulációs játékok hagyományait, főleg a DAMBUSTERS című programhoz hasonlítható. A játék folyamán egyszemélyben testesítjük meg az US Navy (Egyesült Allamok Haditengerészete) egyik tengeralattjárójának személyzetét. A tengeralattjáróval a II. Világháború ázsiai hadszínterén tevékenykedünk, célunk a japán hajókonvojok megtámadása és a hajók elsüllyesztése.

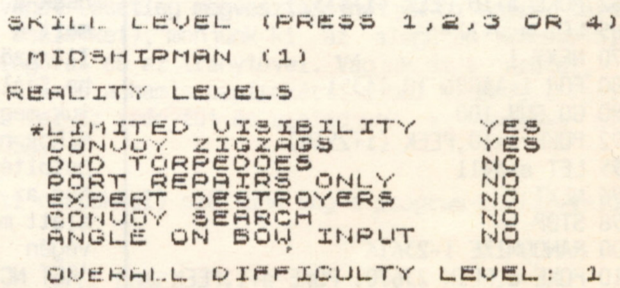


A játék a betöltődés után egy hárompontos menüvel (SCENARIO SELECTION) jelentkezik be. Az első pont (TORPEDO/GUN PRACTICE) az abszolút kezdőknek ajánlott, ebben négy lehorgonyzott teherhajót kell elsüllyeszteni (iszonyú nehéz feladat...), közben pedig be lehet gyakorolni a tengeralattjáró irányítását, a műszerek kezelését.
A második opciót (CONVOY ACTIONS) választva hajókonvojokat támadhatunk meg. Újabb menü tűnik fel, ahol a beosztás és a támadás körülményeinek választásával beállíthatjuk a nehézségi fokozatot. A beosztás kiválasztása az 1-4 billentyűk használatával történik:

1. MIDSHIPMAN (tengerész)
2. LIEUTNANT (hadnagy)
3. COMMANDER (parancsnok)
4. CAPTAIN (kapitány)

A REALITY LEVELS felirat alatt lévő körülmények között az 'Y' gombbal fel, 'H' gombbal lefelé léphetünk, az 'M' gombbal állíthatjuk be a megfelelő választ (YES/NO). A következő körülmények között kell választanunk:

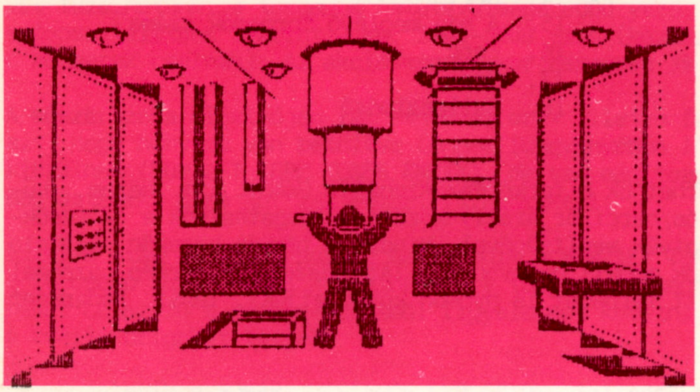
- Korlátozott látási viszonyok
- Cikk-cakkban haladó konvoj
- "Döglött" torpedók
- A javítások csak a kikötőben végezhetőek el
- Több és ügyesebb romboló védelmezi a konvojt
- Az akciót nem a konvoj közelében kezdjük, meg kell keresnünk a hajókat



A képernyő alján az OVERALL DIFFICULTY LEVEL felirat után látható a beállított paraméterekből kialakult nehézségi fokozat.
Ha a nehézségi fokot beállítottuk a SPACE megnyomásával - a változatosság kedvéért - egy újabb menübe lépünk, ahol az 1-6 billentyűk segítségével kiválaszthatjuk, hogy melyik "történelmi" támadást kívánjuk végrehajtani. A megfelelő számbillentyű megnyomása után elkezdődik a játék.

A hajón összesen 6 fülke található, amelyeket a 'CAPS SHIFT' és az '1-6' számbillentyűk együttes megnyomásával tudjuk választani. Az alsó input-sorban minden fülkében a legfontosabb információk láthatóak: a kormányok használatát a tengeralattjáró-

szimbólum mellett és felett kis nyilak jelzik; SPEED (sebesség); DEPTH (mélység); HEAD (a tengeralattjáró orrának irányszöge). A felette levő sorban a hajóra illetve a támadásra vonatkozó információk jelennek meg. A hajónapló használatakor ezek az információk eltűnnek és az irányító szervek sem használhatók.



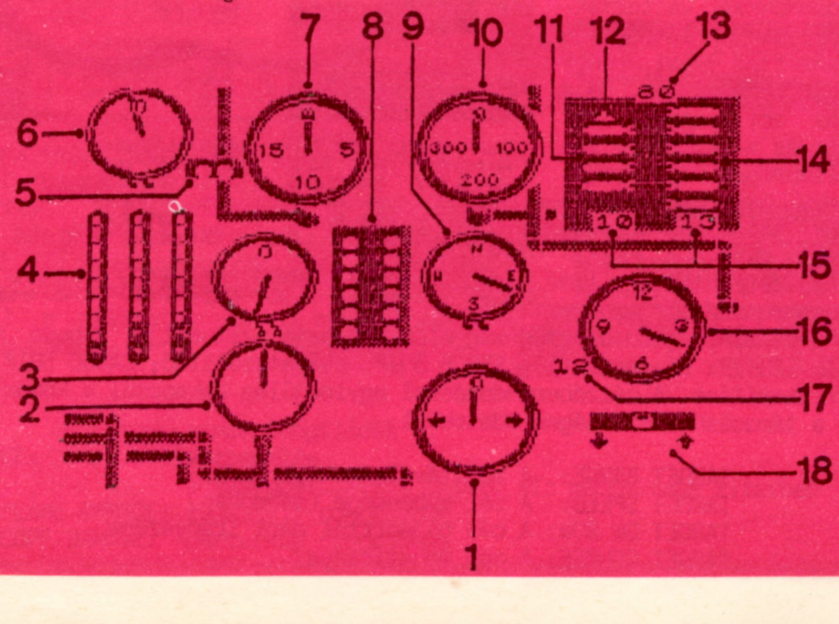
Ha a CAPS SHIFT és a számbillentyűk használata kényelmetlen, a 'SPACE'-szel átválthatunk a vezérlőterembe, ahol a '9-0'(balra a naplóhoz/jobbra a térképhez) és az 'Y-H'(fel a hídra/le a műszerész fülkébe) billentyűkkel mozoghatunk a kívánt pozícióba. A fülke meghatározásához az adott iránybillentyű és az 'M' megnyomása szükséges. Ha az 'M'-et csak önmagában használjuk, a periszkóp ugrik be (feltéve, ha nem vagyunk víz alatt).

A hajó irányítása

Tengeralattjárónkat a 'C' és a 'V' gombok segítségével forgathatjuk balra illetve jobbra. A kormányokat két fokozatban lehet állítani, amit a gép az input-sorban jelez (RUDDER LEFT/RIGHT vagy FULL RUDDER LEFT/RIGHT). A kormányokat az 'ENTER' megnyomásával hozhatjuk egyenesbe (RUDDER AMIDSHIPS). Álló helyzetben is fordulhatunk. Lemerülni a 'D' (DIVE, CLEAR THE BRIDGE), felszínre jönni az 'S' (SURFACING) billentyű megnyomásával tudunk. A megfelelő mélység elérésekor az 'ENTER'-rel állíthatjuk egyenes helyzetbe a kormányokat. Maximum 400 láb mélységig merülhetünk le, ennél többet nem bír ki a hajótest. Ha sekély vízben merülünk le előfordulhat, hogy megfeneklünk a tengerfenéken (WE HAVE AGROUND). Ilyenkor feljebb kell kissé emelkednünk és továbbhaladhatunk.

Sebességünket az '1-5' számbillentyűkkel állíthatjuk be, az alsó sorban megjelenik a ... SPEED AHEAD felirat és a tengeralattjáró felgyorsul az adott fokozathoz tartozó sebességre:'1'- motorok állj; '2'- 1/3 sebesség (5 csomó); '3'- 2/3 sebesség (10 csomó); '4'- TOTAL (15 csomó); '5': FLANK (20 csomó). Ha tolatni akarunk, az 'R' billentyűvel megfordíthatjuk a hajtócsavarok forgási irányát (... SPEED BACK). Újabb 'R' megnyomásra a hajó ismét előre fog haladni.

A 'CAPS SHIFT+4' billentyűk megnyomásával hívhatjuk a műszerész fülkéjét,ahol a következő műszereket láthatjuk:



2.4 SILENT SERVICE

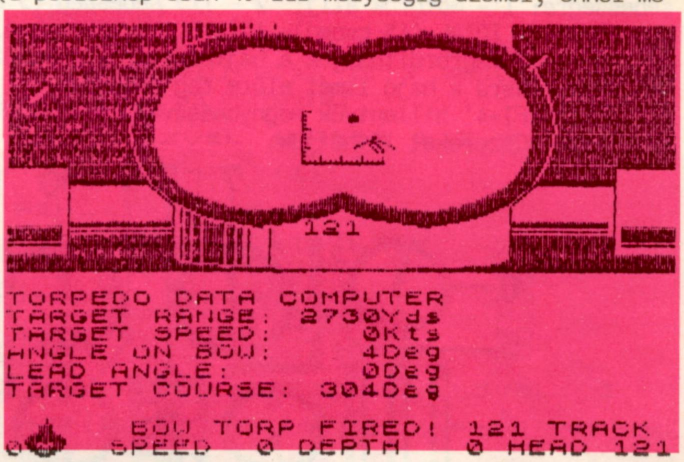
1. Irányjelző műszer
2. A hajótestre ható külső nyomás
3. A generátorok teljesítménye
4. Üzemanyagtartályok és a hőmérő kijelzője
5. Elektro/dieselmotor kijelző
6. Akkumulátor töltésjelző
7. Sebességmérő
8. Kipufogó és szellőzőnyílások (sötét: zárva)
9. Iránytű
10. Mélységmérő
11. Fartorpedók (sötét: kilőtt, üres cső
12. Periszkóp kint/bent
13. Lőszerkészlet az ágyúhoz
14. Orrtorpedók
15. A még meglévő torpedók száma
16. Perc
17. Óra
18. Ballasztmérő

A 'CAPS SHIFT+1' segítségével kérhetjük a térképész szobáját (MAP). Itt három, különböző léptékű térkép (navigációs/harci/őrjárati) között váltogathatunk a 'Z'(csökkentés) és az 'X'(növelés) billentyűk segítségével. Mindig a térkép közepén látható a tengeralattjárónk. A mozgó hajók mögött nyomvonalat láthatunk. A térkép azonnal feltünteti a mozgást (pl. a kilőtt torpedót is nyomon követhetjük), tehát radarként is szolgál.

Harci tevékenység

A 'CAPS SHIFT+2' megnyomásával kapcsolhatunk a parancsnoki hídra (BRIDGE). Itt található a 4 collos fedélzeti löveg, amellyel torpedófogytán folytathatjuk a japán hajók megdorgálását (természetesen csak akkor, ha a felszínen vagyunk). Az ágyút a '9' és a '0' gombokkal forgathatjuk balra illetve jobbra, a célzás távolságát (GUN DEFLECTION) a 'J' billentyűvel csökkenthetjük, a 'K'-val növelhetjük. Az ágyú a 'G' gombbal süthető el, a robbanásokat a horizont felvillanása jelzi (ez nem jelent szükségszerűen találatot). Kezdetben összesen 80 db lőszer áll rendelkezésünkre. Megjegyzendő, hogy az ágyú a periszkóptól is vezérelhető, tehát ezt a helyszínt nem szükséges használnunk.

A 'CAPS SHIFT+3' segítségével hívhatjuk a periszkóphelyiséget (PERISCOPE). Itt találhatóak a torpedók célzóműszerei is. Ha a víz alatt haladunk a 'P' gombbal tudjuk a periszkópot kiengedni/behúzni (a periszkóp csak 45 láb mélységig üzemel, ennél mélyebben már nem használható). Kiengedés után a periszkóp mindig a menetirányba néz, a látóteret vagy a tengeralattjáróval történő fordulással vagy a '9' és a '0' gombokkal változtathatjuk. A periszkóp látómezeje alatt látható szám jelzi, hogy hány fokos irányba nézünk. Ha a periszkóp az orr vagy a far irányába néz és a látótérbe valamilyen ellenséges hajó kerül, automatikusan bekapcsolódik a torpedóvetők célzóberendezése, megjelenik egy skála. A célkeresztben lévő célpontot az 'I' billentyű megnyomásával azonosíthatjuk, a képernyő jobb oldalán megjelenik a hajó sziluettje, típusa (DESTROYER: romboló; OIL TANKER: olajszállító; TROOP SHIP: csapatszállító; CARGO SHIP: teherhajó) és vízkiszorítása. A célzóberendezéssel egyidejűleg bekapcsol a komputer is, ellát minket a támadáshoz szükséges adatokkal:



TARGET RANGE: A cél távolsága (yardban)
TARGET SPEED: A cél sebessége (csomóban)
TARGET ON BOW: A cél hajóorrtól mért irányszöge
TARGET COURSE: A cél menetiránya

A torpedóvető csövek a hajó orrában (BOW TORPS) és farában (AFT TORPS) helyezkednek el. Ha a cél bemérése megtörtént (az ellenség a célzóműszerben van) a 'T' billentyűvel tüzelhetünk. A torpedók hatótávolsága 6000 yard, az ideális tüzelési távolság - tapasztalatunk szerint - kb. 2000-3000 yard (közelről nem lehet pontosan célozni, a műszerek csalnak). Az akció kezdetekor összesen 34 db torpedóval rendelkezünk.

A műszerszoba kijelzőjén láthatjuk az orr- illetve fartorpedók mennyiségét illetve azt, hogy hány db van betöltve (ennek függvényében lőhetünk sorozatot). Ha a csövek kiürülnek (BOW/AFT TORP TUBES EMPTY), de van még torpedónk, a töltés automatikusan megtörténik. Erre kb. 1-2 percet várnunk kell.

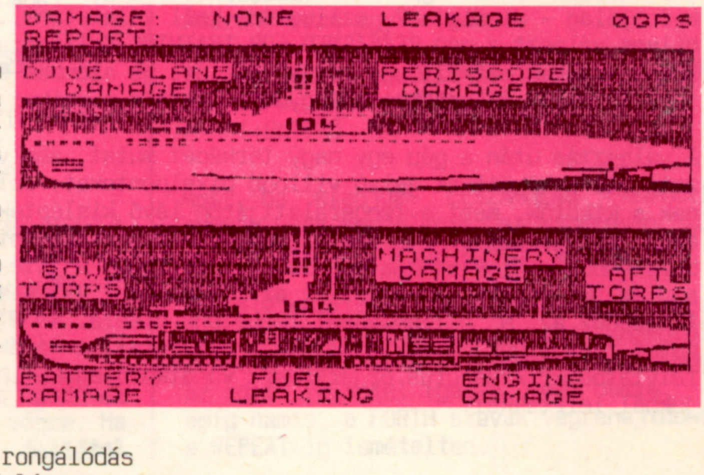
Tüzelés után az input-sorban megjelenik a BOW/AFT TORP FIRED felirat és a cél felé - 24 csomó sebességgel - megindul egy torpedó (a képen is látható). A torpedó haladását célszerű a térkép-szobából figyelemmel követni, mert a periszkópon nem mindig a reális dolgokat látjuk.

Ha megjelenik a SONAR: DISTANCE EXPLOSIONS felirat, az azt jelzi, hogy a torpedónk talált. Pontos találatnál a cél ENEMY SHIP SUNK felirattal hullámsírba merül, kevésbé pontos találatnál a sebessége a felére csökken és újabb torpedót kell beleeresztenünk. Megemlítendő, hogy - valószínűleg programhibából adódóan - ha a már süllyedő hajót újabb találat éri, akkor "megelevenedik".

Ha az összes torpedónk és lőszerünk elfogyott (hajó viszont még van), lophatunk néhány ötletet a japán hadművészetből: kamikaze jelleggel megrohamozhatjuk a célpontot. Ekkor a játék véget ér (RAMMED BY ENEMY SHIP), de a megcélzott hajó még felkerül az általunk elsüllyesztett hajók listájára.

A hajókonvojokat rombolók védelmezik, amelyek általában rossz szemmel nézik tevékenységünket. Ha egy hajó 12 csomós sebességgel közeledik felénk, akkor biztosak lehetünk benne, hogy egy romboló próbál bennünket elhessegetni. Ha ágyúiból nyit ránk tüzet (DESTROYERS FIRING), akkor csekély lemerüléssel és a periszkópon keresztül vezérelt torpedókkal könnyen kivédhetjük, de ha a rombolók bekerítenek (DESTROYERS CLOSING UP) és elkezdik a vizet telehinteni mélységi bombákkal (DEPHT CHARGES DROPPED), jobban tesszük ha behúzzuk a periszkópot és 80-100 láb mélységig merülünk és elhúzzuk a csíkot.

Mint az eddigiekből is kiderül, japán konvojokat megtámadni nem kimondott életbiztosítás, a tengeralattjárónkat több sérülés is érheti, esetleg el is halálozhatunk (YOUR SUB IS LOST AT SEA). A rongálódásokat a 'CAPS SHIFT+5' billentyűkkel előhívható kárjelentési képen (DAMAGE REPORT) vizsgálhatjuk meg (a képen az összes lehetséges látható), a feliratok jelentése a következő:

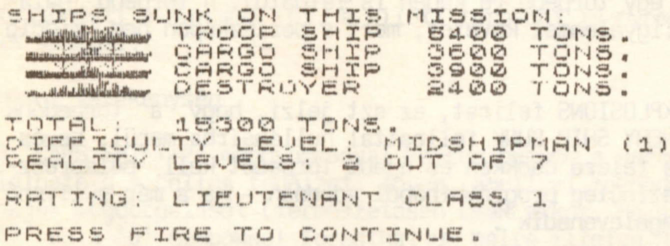


NONE:	nincs rongálódás
LEAKAGE:	vízbeömlés
GPS:	a másodpercenként betörő víz mennyisége literben
DIVE PLANE DAMAGE:	fedélzeti rongálódás
PERISCOPE DAMAGE:	periszkóp rongálódás

2.4 SILENT SERVICE

BOW TORPS:	nem használhatók az orrtorpedók
AFT TORPS:	nem használhatók a fartorpedók
MACHINERY DAMAGE:	gépszerkezet meghibásodás
BATTERY DAMAGE:	elromlottak/kimerültek az elektromotor elemei
FUEL LEAKING:	üzemanyagszivárgás
ENGINE DAMAGE:	meghibásodott a diesel-motor

Ha a nehézségi fokozatok meghatározásánál a PORT REPAIRS ONLY rovatot NO-ra állítottuk be, az egyes (nem végzetes) meghibásodások idővel "kijavítódnak", ellenkező esetben csak halmozódnak egymásra és ha nem érünk el idejében egy kikötőt, elsüllyedésünket fogják eredményezni.



A 'CAPS SHIFT+6' billentyűk megnyomásával hívhatjuk elő a hajónaplót (az irányító szervek nem üzemelnek). Itt a SHIP SUNK ON THIS MISSION felirat alatt láthatjuk az elsüllyesztett hajók sziluettjét, típusát és vízkiszorítását. A TOTAL rovatban szerepel az elsüllyesztett hajók össztonnasúlya, a DIFFICULTY és a REALITY LEVEL a beállított nehézségi fokozatokat jelzik. A RATING után látható, hogy a fokozathoz képest elért eredmény alapján milyen besorolásba javasol bennünket a gép. A tűzgomb megnyomása után visszatérhetünk a kabinba.

Ha sikerült az összes japán hajót elsüllyesztenünk vagy - ne adj'isten - érdemeink elismerése mellett elhaláloztunk, a gép a parancsnoki napló alapján összeállítja a jelentést az akcióról, majd a tűzgomb megnyomása után visszaadja a főmenüt.

Azok, akik a konvojok megtámadásakor aratott számos győzelem birtokában már elég tapasztaltnak érzik magukat, válasszák a főmenü 3. pontját (WAR PATROL). Ezután a gép letesztel bennünket: négy kép közül fel kell ismernünk, hogy melyik ábrázolja a japán rombolók sziluettjét (nekünk nem a gyári, hanem a Rudy&Futuresoft nevű titokzatos emberek által megpiszkált program van meg, ebben - valószínűleg a piszkálásból kifolyólag - mindig a 2. sziluettet kell választani). Ha nem sikerül eltalálnunk a helyes választ a gép REASIGNED FOR TRAINING felirattal visszaküld minket gyakorolni, automatikusan beindít egy konvoj-akciót. Ha azonban sikerült eltalálni, hogy melyik a japán hajó, választhatunk egyet a hatféle amerikai tengeralattjáró őrjárata közül.

A választás után a gép egy nagy térképet mutat, amelyen megnézhetjük az őrjárat körzetét és a kikötőket jelző villogó pixelpontokat. Jól figyeljük meg, hogy merre vannak a kikötők, mert a tengeralattjárón lévő kisléptékű térképek segítségével - az akció befejezése után - nem nagyon tudunk visszatérni (ugyanis vissza kell jönni a bevetésből). Visszatérés közben hasznos lehet a játék "időfelgyorsító" szolgáltatása: az 'F' gomb nyomogatásával az idő múlását 2-3-4-szeresére növelhetjük, az eredeti időhöz az 'N' billentyűvel térhetünk vissza. A 'W' billentyű megnyomása után szünetet tarthatunk, áttekinthetjük a helyzetet. A játék bármely billentyű megnyomására folytatódik. A háborús őrjáratokra vonatkozó egyéb információk megegyeznek a konvojoknál leírtakkal.

A szimulációs és a harci játékok rajongói megérdemelten kedvelik a SILENT SERVICE-t, ez valóban az egyik legjobb és legérdekesebb szimuláció. Azoknak, akik csak ezután fognak tengeralatti küldetésre indulni, a - fentebb leírtakon kívül - egyik barátunk kedvenc mondását tudjuk útravalóul adni: "Áldjon meg titeket a Nemo kapitány!"

A második verzió általános formája:

: DEFINÍCIÓ határ start DO.....'FORTH szavak' növelés + LOOP ; (+ENTER)

Az előzőhöz képest annyi a változás, hogy a start növekményét (vagy csökkenését) mi adhatjuk meg. Pl:

: MINTA2 6 0 DO . " DEMO " CR 2 +LOOP ; (+ENTER),

majd MINTA2 (+ENTER) hatására csak háromszor fog megjelenni egymás alatt a felirat.
Gyakran szükség lehet rá, hogy az index értékét átvigyük a verembe, ilyen adatmozgatásra adnak lehetőséget a következő parancsok:

I : a visszatérési verem az adatverembe másolódik

J : az ágyazott LOOP indexe a verembe másolódik

K : a kétszeresen ágyazott LOOP indexe a verembe másolódik.

Erre is nézzünk példát:

: MINTA 3 0 DO 3 0 DO 3 0 DO K J I ... CR LOOP LOOP LOOP ; (+ENTER),

majd MINTA3 (+ENTER)

Eredménye: 1 1 1
1 1 2
1 1 3 (és folytathatnánk)

FELTÉTELES ELÁGAZTATÁSOK

Ezeket is csak kettőspont definíciókon belül alkalmazhatjuk.
Két általános formája a következő:

: DEFINÍCIÓ IF(igaz)...'FORTH szavak' ...ENDIF ; (+ENTER), ill.
: DEFINÍCIÓ IF(igaz)...'FORTH szavak' ...ELSE (hamis)...' FORTH szavak '... ENDIF ; (+ENTER)

A TOS tartalmától függ, hogy a gép az igaz, vagy a hamis részt hajtja végre. Ha a TOS igaz, vagyis az IF feltétel eredménye nem zérus, akkor az igaz rész hajtódik végre, egyébként ha a TOS hamis, vagyis az eredmény zérus, akkor a hamis rész kerül végrehajtásra.

Az ELSE nem feltétlenül kötelező, ha hiányzik, a végrehajtás az ENDIF-nél folytatódik.
A felhasználható matematikai műveletek a következők:

0< : Ha TOS<0, akkor az eredmény igaz
0= : Ha TOS=0, akkor az eredmény igaz
< : Ha TOS2<TOS, akkor az eredmény igaz
Pl. 30 20 < . (+ENTER) eredménye zérus

> : Ha TOS2 TOS, akkor az eredmény igaz
= : Az eredmény csak akkor igaz, ha TOS2=TOS

TOVÁBBI CIKLUS-SZERKEZETEK

A WL A DO...LOOP cikluson kívül ismeri a BEGIN...UNTIL szerkezetet is.
A ciklusok két főbb általános formája a következő:

: DEFINÍCIÓ BEGIN...'FORTH szavak'... UNTIL ; (+ENTER), ill.
: DEFINÍCIÓ BEGIN...'FORTH szavak'... WHILE...'FORTH szavak'...REPEAT ; (+ENTER)

Ha az első verziót alkalmazzuk, úgy egészen UNTIL-ig a TOS-t vizsgálja a ciklus. Amíg hamis a feltétel, addig a végrehajtás megismétlődik a BEGIN-től. Amikor a feltétel igaz lesz, akkor lépünk ki a ciklusból.
Nézzünk erre egy példát:

: LESZAMOLAS DECIMAL 20 BEGIN 1- DUP DUP . CR 0= UNTIL . " KESZ " ; (+ENTER)

Most, ha kiadjuk: LESZAMOLAS (+ENTER), egymás után megjelenik a képernyőn 20, 19, 18...2,1,0, KESZ egymás alatt.

A másik verzió lehetőséget biztosít arra, hogy a ciklusból menet közben (középről) lépjünk ki. A WHILE vizsgálja meg, hogy az adott feltétel igaz-e, vagy hamis, és amíg hamis, a FORTH szavak végrehajtódnak a REPEAT-ig ismételten.

Erre is nézzünk egy példát, írassuk ki az egész számok köbeit 1-től folyamatosan egyesével, mindaddig, amíg a köbérték el nem éri az 1800-at, ezesetben "MINDEN KESZ" felirattal álljon meg a program futása.

```
: PROGRAM DECIMAL 0 BEGIN 1+ (+ENTER)
DUP DUP DUP DUP (+ENTER) 1800 < WHILE
. " A KOBE " (+ENTER) SWAP . " -NEK "
. CR REPEAT (+ENTER) DROP DROP DROP
" MINDEN KESZ! " CR ; (+ENTER)
```

Ezután adjuk meg: PROGRAM (+ENTER) és nézzük meg az eredményt!

Végül ebben a csoportban még egy szerkezetet ismerünk meg, ez a CASE... OF szerkezet.
Általános formája a következő:

```
: DEFINÍCIÓ CASE N1 OF (FORTH szó)
ENDIF N2 OF (FORTH szó) ENDIF....END
CASE ; (+ENTER)
```

A működés szemléltetésére nézzünk meg egy példát:

```
: PROGRAM CASE 1 OF . " ELSO ESET "
ENDIF 2 OF . " MASODIK ESET " ENDIF 3
OF . " HARMADIK ESET " ENDIF ENDCASE
; (+ENTER)
```

Ezután pedig adjuk meg a következőt:

```
1 PROGRAM CR 2 PROGRAM CR 3 PROGRAM
CR (+ENTER)
```

KONSTANSOK és VÁLTOZÓK

A konstans- és a változó-kezelési mechanizmus között az az alapvető különbség, hogy, ha egy konstanst hívunk, az automatikusan a TOS-ba kerül, míg egy változó hívása esetén a változó címe tárolódik el a TOS-ban.
A konstans általános definíciója:

```
ÉRTÉK CONSTANT NÉV
```

A változó általános definíciója:

```
ÉRTÉK VARIABLE NÉV
```

Nézzünk néhány műveleti példát:

```
23 CONSTANT S 500 CONSTANT P (+ENTER)
202 VARIABLE M (+ENTER)
20 VARIABLE X (+ENTER)
```

S P + . (+ENTER) eredménye S és P összege lesz, tehát 523.
M. (+ENTER) eredménye nem a változó értéke, hanem annak címe lesz.
X@. (+ENTER) eredménye X változó értéke lesz.
PX ! (+ENTER) eredményeképpen X változó felveszi P értékét.
4M ! (+ENTER) eredményeképpen az M változó értéke 4 lesz.

ÁLTALÁNOSAN HASZNÁLT FORTH SZAVAK

LIST: A TOS-ban megadott konstans szerinti képernyő kerül kilistázásra. Pl. 4 LIST (+ENTER) a 4. számú képernyőt fogja kilistázni.

FORGET: Segítségével törölhetünk a WL. FORTH szótárából.
Figyelem! a FORGET törli a megadott szót, és mindazokat a szavakat is, amelyeket a megadott szó után vittünk be a memóriába!

LOAD: A WL. szótárába beteszi azokat a szavakat, amelyeket az EDITOR-ral szerkesztettünk meg.

VLIST: Paraméter nélkül kilistázza a memóriában levő FORTH szótár szavait. A listázás Break-kel megszakítható.

AZ EDITOR (szerkesztő) HASZNÁLATA

Fontos! Egy sor maximálisan 64 karakterből állhat, ennél többet a WL. figyelmen kívül hagy!
A gép a szerkesztés megkönnyítésére a szövegeket 512 byte-os egységekbe szervezi, azaz 8 db 64 karakteres sorba. Az elmentés szerkesztés után azonnal elvégezhető.
A szöveget mindig a memória HEX C000-F000 címei között találjuk, ebből adódóan 1-23 képernyő terjedelmű szöveg szerkeszthető.

AZ EDITOR UTASÍTÁSAI

H: (HOLD) Az ideiglenes memóriaterületen (PAD) tartja azt a szövegrészt, ahova a TOS mutat.

S: (SPREAD) A megadott számú sort feltölti üres karakterekkel, a soron következő sorokat eggyel lefelé lépteti, így pl. '5 S' a felső 5 sort törli, a 6. sortól pedig léptet egy sort lefelé.

D: (DELETE) Törli a TOS által meghatározott sort. Az alatta levő sorok eggyel feljebb lépnek. A törölt sor még ezután a PAD-ben megtalálható, ha esetleg később szükség lenne rá.

E: (ERASE) Törli a TOS által meghatározott sort, pontosabban SPACE -ekkel tölti fel.

RE: (REPLACE) A TOS által meghatározott sort felcseréli a PAD-ban eltárolt sorral.

P: (PUT) A soron következő szöveget a TOS szerinti sorba írja, felülírva annak tartalmát.

INS: (INSERT) A PAD-ben tárolt szöveget az aktuális sorhoz inzertálja, az adott és alatta elhelyezkedő sorok eggyel lejjebb lépnek.

EDIT: A Spectrum BASIC EDIT-tel analóg.

CLEAR: A TOS által meghatározott képernyőt törli, és ez lesz az aktuális képernyő.

WHERE: Ez akkor aktivizálható, ha a szöveg betöltésekor hiba fordul elő. Ekkor a parancs hatására a gép kijelzi a hibás sort, és az kijavítható.

LOAD: A veremben (TOS) megadott számú képernyőtől betölti a szöveget annak aljáig. Folyamatos szöveg esetén betöltéskor a képernyő végén egy "→" jel kell, hogy megjelenjen. Ez azt jelenti, hogy "S" kivételével bármelyik billentyűt nyomjuk meg, folytathatjuk a betöltést, viszont "S" megnyomása esetén a töltés befejeződik.

FLUSH: Minden kimentés előtt ezt kell kiadnunk. A kimentés BASIC-ből oldható meg a normál SAVE utasítás segítségével, így a FLUSH után vissza kell térni először is BASIC-be. Ez a PROG (+ENTER) segítségével lehetséges.

Mintapélda a szerkesztésre:

- Töröljük a képernyőt: 9 CLEAR (+ENTER)Ez a 9. számú képernyőt aktuálissá teszi.

- Írjuk be a 0. sort: 0 EDIT (+ENTER) A 0. sor szerkesztésre kész, írjuk be: ' MOST MEGTUDJUK HOGY KELL'

- Írjuk be az 1. sort: 1 EDIT (+ENTER), majd
'BEIRNI AZ EGYES SORBA'
Ezután a 9 LIST (+ENTER) segítségével megjelenik:

```
0 MOST MEGTUDJUK HOGY KELL
1 BEIRNI AZ EGYES SORBA
```

Az 1. sor megváltoztatásához egyszerűen gépeljük be: 1 EDIT (+ENTER), és a szerkesztő mód felhasználásával szúrjunk be pl. egy betűt, majd ismét adjuk meg 9 LIST (+ENTER) és nézzük meg az eredményt.

Szöveg kinyomtatása:

A Printer üzemmód bekapcsolása a PRT-OFF parancsokkal lehetséges. Nyomtatáshoz érdemes definíciót alkotnunk, hogy egy szó bevitelével végrehajthassuk a műveletet.

```
Pl. : SLIST PRT-ON 1+ SWAP DO 1 LIST CR
    LOOP PRT-OFF ; (+ENTER)
```

Ha a 7-9 képernyőket ki akarjuk listázni, ez a következőképpen lehetséges:

- a képernyőre: 7 9 LIST (+ENTER)
- a printerre: 7 9 SLIST (+ENTER)

A FORTH HIBAÜZENETEI

A hibaüzenetek általános megjelenési formája a következő:

'HIBÁS SZÓ' ? MSG# szám

Ebből a szerkezetből a legfontosabb a '#szám' jelzés, így most ezeket tekintsük át:

#0 - a szó nem található, vagy lehetetlen a numerikus átalakítás
#1 - üres a verem, de adatokat akarunk kivenni, vagy több adatot akarunk kivenni, mint amennyi a veremben van
#2 - a szótár megtelt
#4 - egy már meglevő szót újradefiniálunk
#6 - nemlétező képernyőt akarunk listázni
#9 - sprite területen nem lehet törölni kevesebb, mint 2 byte-tal (ld. később)
#10 - az IDEAL szó nemlétező sprite-ra hivatkozik, vagy ISPRITE segítségével be akarunk szúrni egy sprite-ot, de a sorszám foglalt (ld. később)
#17 - direkt módban akarunk olyan szót felhasználni, amely csak szerkesztéskor használható
#18 - definícióban akarunk olyan szót felhasználni, amely csak direkt módon alkalmazható
#19 - a definícióban a feltételes szerkezet egyik utasításának hiányzik a párja
#20 - a kettőspont definíció nem megfelelően fejeződik be
#21 - törölni akartunk a szótárból olyan szót, amit nem lehet
#22 - a "→" jel helytelen felhasználása szövegképernyők betöltésekor.
#23 - nemlétező sort akarunk szerkeszteni

Az "IDEAL" nyelv

ALAPFOGALMAK

Az IDEAL nyelvet azért tervezték, hogy a felhasználó számára megkönnyítse a sprite-ok és a képernyőadatok kezelését. Több, mint 100 - a FORTH-tal is keverhető - új parancsával dinamikus segítséget nyújt bármely programozónak, aki maximálisan ki akarja használni a ZX Spectrum grafikai képességeit. A FORTH ismertetésénél megláthattuk, hogy - a meglévők felhasználásával alkotott - új parancsok mennyire kényelmessé teszik a programfejlesztést.

A későbbiek könnyebb megértése végett mindenekelőtt tisztázzunk néhány alapfogalmat:

-sprite: software úton ellenőrizhető önálló kép (karakter). A WHITE LIGHTNING egyszerre akár 255 db-ot is képes tárolni.

-ablak: a képernyő önálló része (" képernyő a képernyőben"), amelyet négy - karakterekben értelmezett -változó határoz meg: COL (oszlop, 0-31); ROW (sor, 0-23); HGT (magasság, 1-24); LEN (hossz, 1-32). A COL és a ROW az ablak bal felső karakterének koordinátáit, a HGT és a LEN pedig az ablak területét határozza meg. A képernyő bal felső koordinátái:ROW 0 és COL 0.

-sprite-ablak: egy sprite önálló része, amelyet négy - az ablakoknál megismertekhez hasonló - változó határoz meg: a SCOL, a SROW, a HGT és a LEN. A SCOL és a SROW a sprite-ablak bal felső sarkát, a HGT és a LEN pedig a sprite-ablak méretét határozzák meg.

-sprite-terület: a memória azon része, ahol a sprite-ok találhatóak. Az SPST tartalmazza a sprite-terület kezdő-, az SPND pedig a végcímét.

Megjegyeznénk, hogy az SPND nem lehet nagyob FFF0h-nál, ha a háttérben a gép rutinokat hajt végre.

-pixel-adatok: egy karakter(blokk)ra vonatkozó adatok összesége. A ZX Spectrumon a karakterek mérete 8x8 pixel. A pixel egy INK vagy PAPER színű képpont.

-attributum-adatok: az egyes karakterek színeit meghatározó byte-ok összessége. A pixel- és attributum-adatokra, mint külön jellemzőkre hivatkozunk.

Tape Copier-7 (LERM)

A TC-6-hoz hasonlóan ez is főként speciális, védett programok másolását teszi lehetővé.
A TC-7-tel megtörhető védelmek:
- nagyon hosszú blokkok
- JERKY (szaggatott) TONE
- TURBO-s programok

4.1 NORMÁL MÁSOLÁS

A normál másolás még "gyermetegebb"-nek tűnik, mint a TC-6 esetében, mivel a '82-es év első másolóinak (pl. Special Copy) szintjén, egyszerre csak egy blokk betöltését és kimentését teszi lehetővé. 'SHIFT+L' segítségével fejléces blokk, 'h' segítségével fejléc nélküli blokk tölthető be. Egy blokk betöltése után azt az 's' billentyűvel menthetjük ki, akár többször is egymás után. Ezzel a módszerrel max. 38590 byte másolható. Ha nem vagyunk biztosak a kód hoszában, nullázzuk le a magnetofont, majd a 'c' billentyű segítségével számoltassuk le a byteok számát. Ezután visszatekerve a szalagot, eldönthetjük, hogy milyen módszerrel másoljuk a programot. A számláló mód (c) turbósított programok esetében is alkalmazható, JERKY fejléceknél viszont nem.
A programból a 'Q' (SHIFT+Q) megnyomásával léphetünk ki.
Az 'L' nem "eszi meg" a hamis fejlécet, de 'h'-val be tudjuk tölteni a fejlécet és a kódot külön-külön.

4.2 'B' MÓD/1

Lehetővé teszi turbósított programok, rövid idejű TONE-ok és hosszú programok másolását.
Az egyes funkciók alkalmazása előtt nem árt tudni a másolandó blokkok pontos hosszát, erre ad lehetőséget a byte számlálás a 'c' segítségével.
A B-módban 12 billentyű-funkciót helyeztek el (a menü a képernyőn nem jelenik meg):
- 'L' (SHIFT L) max. 48300 byte betöltése (a kép jobbra tolódik)
- 'm' 48300-50000 byte betöltése (a kép zavaros lesz)
- 'r' 50K-nál hosszabb programok másolása
- 'j' 48,3K-nál hosszabb, normál sebességű programok másolása, ahol a betöltés nem a képernyőn kezdődik, így a kép nem romlik el
- '3' 48,3K-nál hosszabb, normál sebességű programok másolása, ahol a betöltés 16384-nél kezdődik, de a kép nem romlik el

4.3 'B' MÓD/2

- 'c' blokkok hosszának számolása byteonként
- 's' betöltött programok kimentése. A kimentés többször is megismételhető, kivéve, ha a programot a 'j', vagy '3' funkcióval töltöttük be
  A turbós programot turbósan is fogja kimenteni
- '0' (zérus) a turbósként betöltött programot normál sebességgel menti el
- '6' a baud sebességének beszabályozására szolgál
- 'p' speciális elmentés. Néha előfordul, hogy 'm'-mel kéne tölteni, de a kimentés hibás. Ez esetben töltsünk az 'L' billentyű segítségével, majd a kimentést 'p'-vel végezzük
- 'q' új program indítása
- 'a' képernyő törlése, alapállapot előidézése a B megnyomása előtti pillanatnak megfelelően.

4.4 'J' MÓD

Lehetővé teszi JERKY (szaggatott) TONEnal ellátott programrészek betöltését.
A J-módban 7 funkció áll rendelkezésre (a menü a képernyőn nem jelenik meg):
- 'b' a JERKY (szaggatás-változás mértéke) lemérése
- 'L' 1. verzió JERKY TONE blokk betöltése
- 'm' 2. verzió JERKY TONE blokk betöltése

A megfelelő verzió a mérés eredménye alapján választható ki. Ha 'L' után a színek nem megfelelőek, 'm'-mel kell tölteni.
- 's' JERKY blokk kimentése a JERK-kel együtt
- '0' JERKY blokk kimentése normál TONE-nal és normál sebességgel
- 'a' alap módra való visszatérés
- 'q' a program megszakítása

5. LEONARDO

A Leonardo grafikai programcsomag széleskörű rajzolási lehetőséget biztosít számunkra. Bár kezelése kezdők számára nehézkes lehet, mégis érdemes vele megismerkedni.
A programfile-ok a gyári kazettán a következőképpen helyezkednek el:

```
pr: leonardo     (  216 byte)
by: ls           ( 6912 byte)
by: lc           (21904 byte)
pr: leonardo     (18232 byte)
-------------------------------
pr: USER         (  812 byte)
by: USER mc      (  856 byte)
```

Betöltés után célszerű a számunkra megfelelő joystick-et kiválasztani, mert ez megkönnyíti további tevékenységeinket.

A LEONARDO jelző rendszer:

A képernyő bal felső sarkában látható színcsík a meghatározó:

- zöld: parancskész állapot
- sárga: végrehajtás, nem lehet megszakítani
- bíbor: végrehajtás, megszakítható

A joystick kiválasztása után a FŐMENÜ-be kerülünk, ahonnan 3 választási lehetőségünk adódik:

1. grafikai munka
2. grafika kimentése
3. grafika betöltése

GRAFIKAI MUNKA

A képernyőn két kurzor van, az egyik látható, és ez mozgatható a joystick-kel, míg a másik a LAST PLOT, az ún. segédkurzor.
A továbbiakban az egyes billentyűkkel elérhető funkciókat ismertetjük:

- Ablakozás:

'2' ablak bal alsó sarkának beállítása
'3' ablak jobb alsó sarkának beállítása
'4' ablakhatár ki/bekapcsolása

- Információs ablak:

'CS+2' ablak kijelzése a bal felső sarokban
'CS+3' ablak kijelzése a bal alsó sarokban
'CS+4' ablak eltüntetése

- Rajzoló üzemmód:

'P' PLOT mód (normál rajzolás)
'O' UNPLOT mód (radírozó rajzolás)
'CS+P' INVERZ mód, de a kurzor nem veszti el a PLOT módot
'M' ecset méretének beállítása
'CS+M' ecset méretének alapállapota (1,1)
'CS+X' rajzolási határ aktivizálása (ha rajzoláskor az ecset kigyújtott pixelt talál, befejezi a rajzolást)

- Lépésköz funkciók:

'I' lépésköz beállítása
'CS+I' lépésköz visszaállítása (1,1)

Kurzor funkciók:

'K' kurzor beállítása (x,y)
'CS+K' kurzor középre állítása
'CS+SPACE' kurzor a bal felső sarokba
'SPACE' a kurzor kijelzése
'Z' - a LAST PLOT beállítása (x,y)
'CS+Z' a LAST PLOT ráállítása a kurzorra
'ENTER' aktuális kurzor mód visszaállítása

Egyenes rajzolása:

'Q' vonal húzása a kurzor és a LAST PLOT közé (LAST PLOT a kurzorra lép)
'CS+Q' vonal húzása a kurzor és a LAST PLOT közé (LAST PLOT helyben marad)
'W' vonal húzása a kurzortól balra/jobbra
'CS+N' vonal húzása a kurzortól felfelé/lefelé
'R' vonalhatár beállítása (a kurzor tovább nem rajzol)
'CS+R' a rajzolás irányának beállítása oda/vissza (a kurzor és a LAST PLOT között)
'F' szaggatott vonal rajzolásához a jel/space arány beállítása
'CS+F' a jel/space arány visszaállítása (1,0)
'CS+C' vonal mód beállítása

Görbék rajzolása:

'A' kör rajzolása (radius=sugár, értékét meg kell adnunk)
'CS+A' kör rajzolása (sugár=kurzor és LAST PLOT távolsága)
'S' ív rajzolása (Arc angle=középponti szög; 90=negyed kör)

'CS+S' ellipszis rajzolása
Inputok: függőleges sugár, vízszintes sugár, forgatás értéke kezdeti szög, elfordítás

Egyéb funkciók:

'CS+SS' képernyő törlése
'U' képernyő invertálása (ablak esetén választhatunk invertálás, törlés esetleg kitöltés között)
'CS+U' Fill mód, zárt területek kitöltése
'J' SCROLL mód - pixel módban pixeleket, karakter módban az attributumokat scrollozza.

Az első választásra 'W' esetén teljes képernyőre, 'S' esetén az ablakra hajtja végre. Ezután (L,R,U,D) a megfelelő irány kiválasztása történik, majd bekéri a gép a lépések számát.
'CS+J' Ismétlődő mintákat generál a képernyőre az UDG készletből. Az input a 'J'-nél megismertek szerint követhető.
'V' Fényes/normál "képernyő sakktábla-háló" ki/bekapcsolása
'CS+V' Fekete-fehér "képernyő sakktáblaháló" ki/bekapcsolása

Memóriaműveletek

A Leonardo két képernyőt kezel egyszerre, az egyiket a szemünk előtt látjuk, a másik hátul található a memóriában (44417-51328). Ezen kívül van egy főtároló (51330-57526), amely tartalmazhat UDG-ket, képelemeket, de egy teljes képernyőt is.

'L' karakterblokk kiíratása (csak karakter kurzor módban)
Itt végezhetők el az UDG műveletek.
'H' az aktuális kép bemásolása a háttértárolóba (a felülírást leellenőrzi)
'Y' visszamenti a képet a háttértárolóból
'CS+H' a képet attributumai nélkül a főtárolóba menti
'CS+Y' a képet visszahívja a főtárolóból és 'A'-(OR); 'C'-(AND); 'D'-(XOR) művelettel egyezteti a két képet
'D' egyszeri megnyomásra ablak bal alsó sarkának beállítása, másodszori megnyomásra ablak jobb felső sarkának beállítása, harmadszori megnyomásra terület eltárolása
'CS+D' Ugyanaz, mint az előző, csak a terület nagyítva (x,y) másolódik át
'G' képelem elmentése
'CS+T' képelem törlése
'N' tárolt képelem megjelenési mód beállítása
'CS+G' képelem visszahívása
'L' képelem ráillesztése a kurzorra (képelem mint ecset)
'CS+E' alak definiálása (10 koordinátapárral)
0-9 - koordinátapárok kiválasztása
ENTER - visszatérés
'J' kezdeti értékek inicializálása
'K' rajzasztal megtekintése
'CS+L' az alakot a kurzorra helyezi (alak mint ecset)

Programozott rajzolás:

'E' belépés a programozott rajzolás asztalához
'1' lépték; '2' lépték követés; '3' START; '4' összekötés; '5' elmozdítás; '6' forgatás; '8' forgatásnövelés; 'A' képadatok; 'B' koordináták; 'C' ecset mód; 'H' paraméter-visszaállítás; 'I' alak asztal; 'D' vonal ; 'E' alak; 'F' kép; 'G' ecset

Színezés:

'B' karakter-mód váltó
'C' színbeállítás
'W' színek aktivizálása
'A' színváltás
'S' színcsere
'F' a színező ablak másolása
'P' kiírás beállítása
'O' kiírás
'C' visszatérés a pixel-kurzor módba

Perifériaműveletek

'C' Hard-copy printerre

Főmenü meghívása:

'X' billentyűvel lehetséges

GRAFIKA ELMENTÉSE (Főmenüből '2' megnyomása)

'1' képernyő elmentése
'2' főtároló elmentése
'0' visszatérés a főmenübe

GRAFIKA BETÖLTÉSE (Főmenüből '3' megnyomása)

'0' visszatérés a főmenübe
A további négy opció pedig a betölteni kívánt adatok típusára vonatkozik.

Menet közben előfordulhat, hogy leáll a program futása. Ezesetben ne 'RUN'-nal indítsuk a programot, hanem a GO TO 9999 utasítás segítségével. (Ne lepődjünk meg, a 9999-es sorban RANDOMIZE USR 0-t olvashatunk!)

LEONARDO User

A USER program öt felhasználói rutint tartalmaz, melyek segítségével lehetőség nyílik kapcsolat létesítésére a saját program és a USER rutinok között. Ezek a rutinok a grafikai adatok kezelését hivatottak megkönnyíteni.

1. szubrutin:

Az egész képernyőnek, vagy egy részének mozgatására szolgál, természetesen a képernyőket előre be kell tölteni a memóriába.
A rutint a 9900-as sorban hívhatjuk meg szubrutinként (GO SUB 9900), de előtte be kell állítani 3 BASIC változót.
MOmód száma (jelzi a mozgatható terület elhelyezkedését)

mód-szám	Felh. graf. tároló	Képernyő
0	egész	egész
1	felső 2/3	felső 2/3
2	felső 2/3	alsó 2/3
3	alsó 2/3	felső 2/3
4	alsó 2/3	alsó 2/3
5	felső 1/3	felső 1/3
6	felső 1/3	középső 1/3
7	felső 1/3	alsó 1/3
8	középső 1/3	felső 1/3
9	középső 1/3	középső 1/3
10	középső 1/3	alsó 1/3
11	alsó 1/3	felső 1/3
12	alsó 1/3	középső 1/3
13	alsó 1/3	alsó 1/3

TY az átvitel típusa

Átvitel típusa -irány

0- Pixelek és színek-képernyőbe tárol
1- Csak pixelek -képernyőbe tárol
2- Csak színek -képernyőbe tárol
3- Pixelek és színek-képernyőről tárolóba
4- Csak pixelek -képernyőről tárolóba
5- Csak színek -képernyőről tárolóba
6- MERGE pixelek OR -képernyőbe tárol
7- MERGE pixelek AND-képernyőbe tárol
8- MERGE pixelek XOR-képernyőbe tárol

PIC: az UDG-k, ill. a grafika kezdőcíme.

2. szubrutin

Ennek segítségével tudjuk előhívni a tárolóból a képelemeket. A szubrutin meghívására a 9905. sorban van lehetőség, de itt is előtte néhány változót be kell állítani.

EN: a képelem-szám
EX és EY: a képelem koordináták pozíciója
EM: az elem visszaállítási módja
'0'-normál; '1'-OR; '2'-AND; '3'-XOR
EB: az elem jelzőbitje (ld. ecsethatás)
EL: a felhasználói grafika tárolójának címe

3. és 4. szubrutin

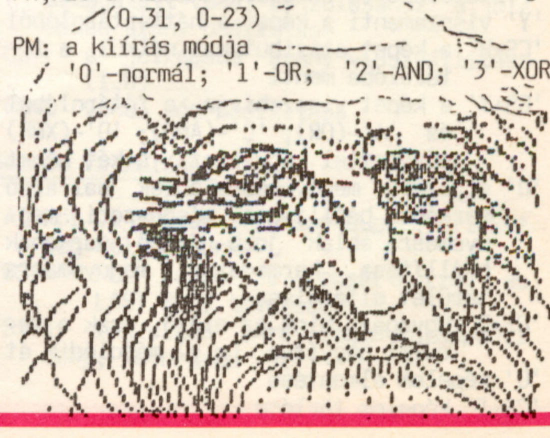
Mindkettő egy-egy új karakterkészlet használatát teszi lehetővé.
A 3. rutint a 9910-es sorban hívhatjuk meg, változói:
CA: a karakterkészlet címe a felhasználói programban
CN: A felhasználói karakterkészletben a SPACE karakter száma

A 4.rutin a 9915-ös sorban hívható, BASIC változót nem igényel, ez állítja vissza az eredeti karakterkészletet!

5. szubrutin

Az UDG blokkok kiíratását oldja meg. Meghívása a 9920-as sorszámon lehetséges, és 7 változót igényel:

PA: az UDG file 1. karakterének címe
PN: az első blokkba írandó karakterek száma (1-774)
PX,PY: a karakterblokk mérete(1-32, 1-24)
PR,PL: a kiírás kezdőértékének pozíciója (0-31, 0-23)
PM: a kiírás módja
'0'-normál; '1'-OR; '2'-AND; '3'-XOR



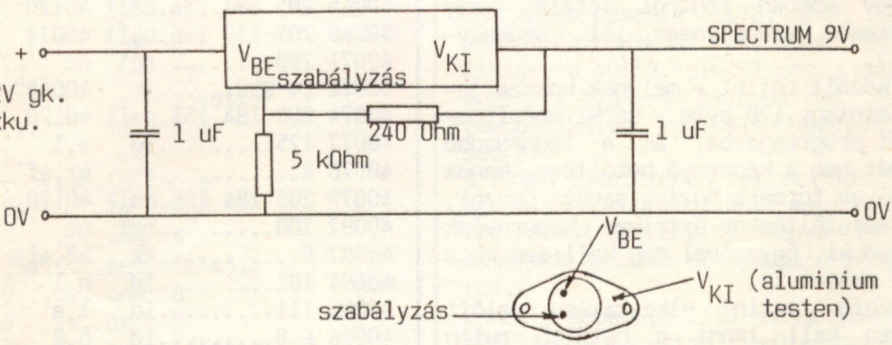
6. HARDWARE ÖTLETEK

6.1 A SPECTRUM AZ AUTÓBAN

Ez a lehetőség mindenkinek adott, akinek 12 V-os akkumulátora van gépkocsijában, vagy lakókocsijában 12 V-os generátorral rendelkezik. A Spectrum működéséhez 9 V; 1,2 A értékű egyenáramra van szükség. A 9 V előállításához két egyszerűbb módszert ismertetünk:

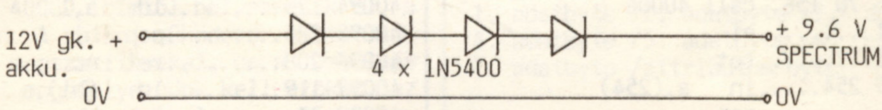
a.) <u>317 K feszültségszabályozóval:</u>

A 317 K feszültségszabályozó lehetővé teszi 1,5 A áram mellett 1,2-37 V határok közötti feszültség szabályozását.
Az áramköri vázlat a következő:



b.) <u>Diódák segítségével:</u>

4 db 1N5400 dióda segítségével a 12 V feszültség 9,5 V-ra osztható le, amelynek áramköri vázlata igen egyszerű:

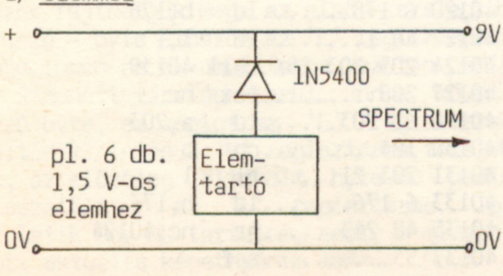


Egy fontos probléma azonban felvetődik. A TV készülék üzemelését (pl. a JUNOSZTY is felhasználható gépkocsi akkumulátorról) a gépkocsi üzeme (elsősorban gyújtása) erősen zavarja, ezért csak a gépkocsi álló helyzetében, kikapcsolt gyújtás mellett érdemes a gépet használnunk!

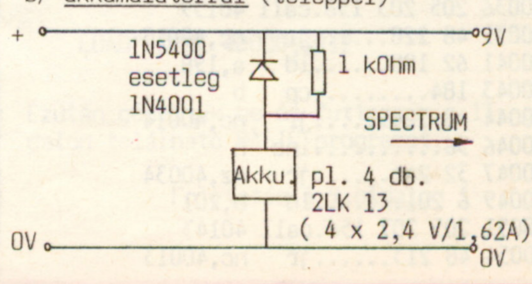
6.2 TARTALÉK-HÁLÓZAT

Előfordulhat, hogy a hálózati feszültség kimarad, és több órás munkánk vész kárba. Ezt elkerülhetjük, ha a mellékelt vázlatoknak megfelelően beépítünk a rendszerbe egy elemtartót (a), vagy egy tartalék telepet (b). Ha feszültség kimaradás lenne, a diódán keresztül még néhány percig megfelelő feszültség látja el a gépet az elemről vagy telepről, így programunk nem vész el, hosszabb kimaradás esetén pedig még van időnk kimenteni bent lévő programunkat (microdrive-ra, vagy elemes magnetofonra).

a/ <u>elemmel</u>



b/ <u>akkumulátorral</u> (teleppel)



Karakteres képernyő-betöltő rutin

Több játék (pl. MIKIE, BAT MAN, ALIEN HIGHWAY, stb.) betöltése közben találkozhatunk ezzel a látványos betöltő rutinnal, amely nem a szokásos formában tölti be a játék képernyőjét, hanem karakterenként vagy sorban fentről lefelé, vagy össze-vissza (szabályosan, ill. szabálytalanul).
Az itt közölt rutint - melynek hossza jelenleg mintegy 178 byte - bárki beépítheti saját programjaiba, ám a legnagyobb problémát nem a képernyő betöltése, hanem annak olyan formára hozása szokta okozni, hogy visszatöltéskor értelmes képernyőnk alakuljon ki. Egyszóval meg kell írnunk a kimentő rutint is.
A kimentő rutin elkészítése előtt tisztában kell lenni a betöltő rutin működésével, így most előbb tekintsük át azt.

```
40000 243.........di
40001 205 70 156..call 40006
40004 251.........ei
40005 201.........ret
40006 219 254.....in   a,(254)
40008 31..........rra
40009 230 32......and  32
40011 79..........ld   c,a
40012 191.........cp   a
40013 192.........ret  nz
40014 205 207 156.call 40143
40017 48 250......jr   nc,40013
40019 33 21 4.....ld   hl,1045
40022 16 254......djnz 40022
40024 43..........dec  hl
40025 124.........ld   a,h
40026 181.........or   l
40027 32 249......jr   nz,40022
40029 205 203 156.call 40139
40032 48 235......jr   nc,40013
40034 6 156.......ld   b,156
40036 205 203 156.call 40139
40039 48 228......jr   nc,40013
40041 62 198......ld   a,198
40043 184.........cp   b
40044 48 224......jr   nc,40014
40046 36..........inc  h
40047 32 241......jr   nz,40034
40049 6 201.......ld   b,201
40051 205 207 156.call 40143
40054 48 213......jr   nc,40013
40056 120.........ld   a,b
40057 254 212.....cp   212
40059 48 244......jr   nc,40049
40061 205 207 156.call 40143
40064 208.........ret  nc
40065 205 184 156.call 40120
40068 205 138 156.call 40074
40071 208.........ret  nc
40072 24 250......jr   40068
40074 205 184 156.call 40120
40077 125.........ld   a,l
40078 8...........ex   af,af'
40079 205 184 156.call 40120
40082 208.........ret  nc
40083 8...........ex   af,af'
40084 101.........ld   h,l
40085 111.........ld   l,a
40086 6 8.........ld   b,8
40088 229.........push hl
40089 197.........push bc
40090 205 184 156.call 40120
40093 193.........pop  bc
40094 125.........ld   a,l
40095 225.........pop  hl
40096 208.........ret  nc
40097 119.........ld   (hl),a
40098 36..........inc  h
40099 16 243......djnz 40088
40101 124.........ld   a,h
40102 214 8.......sub  8
40104 31..........rra
40105 31..........rra
40106 31..........rra
40107 230 3.......and  3
40109 246 88......or   88
40111 103.........ld   h,a
40112 229.........push hl
40113 205 184 156.call 40120
40116 125.........ld   a,l
40117 225.........pop  hl
40118 119.........ld   (hl),a
40119 201.........ret
40120 6 178.......ld   b,178
40122 46 1........ld   l,1
40124 205 203 156.call 40139
40127 208.........ret  nc
40128 62 203......ld   a,203
40130 184.........cp   b
40131 203 21......rl   l
40133 6 176.......ld   b,176
40135 48 243......jr   nc,40124
40137 55..........scf
```

```
40138 201.........ret
40139 205 207 156.call 40143
40142 208.........ret  nc
40143 62 22.......ld   a,22
40145 61..........dec  a
40146 32 253......jr   nz,40145
40148 167.........and  a
40149 4...........inc  b
40150 200.........ret  z
40151 62 127......ld   a,127
40153 219 254.....in   a,(254)
40155 31..........rra
40156 208.........ret  nc
40157 169.........xor  c
40158 230 32......and  32
40160 40 243......jr   z,40149
40162 121.........ld   a,c
40163 47..........cpl
40164 79..........ld   c,a
40165 237 95......ld   a,r
40167 230 7.......and  7
40169 211 254.....out  (254),a
40171 230 0.......and  0
40173 246 8.......or   8
40175 211 254.....out  (254),a
40177 55..........scf
40178 201.........ret
```

A rutin a 40000. címtől helyezkedik el a memóriában. A rutin működése alatt letiltjuk a megszakításokat, de a betöltés után ismét engedélyeznünk kell azokat, ez a mechanizmus játszódik le a rutin elején. A tényleges betöltő rutinba a 40006. címen lépünk be, ez egyébként itt még megegyezik a ROM LOAD rutinjával, annak megfelelően az 1378. címmel, vagyis itt az EAR bit beolvasása történik. A BORDER kezelő utasítások szándékosan maradtak el, betöltéskor ki fog derülni, hogy miért. 40013-tól kezdődik a jelváltás vizsgálata, 40034-től pedig a bevezető jel fogadása. A szinkronizáló impulzus 'ki/be' részeit a 40049. címtől adjuk ki, majd a 40064. címen található RET NC utasítással befejeződik a ROM analógia. 40061-40119-ig tart a program betöltés szempontjából érdekes része. Itt a következőt fejthetjük meg: az első - szalagról érkező - byte először az 'L' regiszterbe, majd innen az akkumulátorba, ezután pedig az átmeneti tárolóba kerül. Jön a következő byte, ez megint az 'L' regiszterbe töltődik. Innen a 'H' regiszterbe töltjük át, az előzőleg eltárolt byte-ot pedig visszatöltjük az 'L' regiszterbe. Így születik egy HL tartalom, amely a mindenkori aktuális képernyőcím lesz, ahová a megfelelő byte-ot töltjük.

No persze az értelmetlen lenne, ha minden egyes képernyőbyte-hoz tartozó cím alsó/felső byte-ját külön betöltenénk, így egy képernyő állomány hossza 20 Kbyte lenne. Ehelyett úgy kell kialakítani a képernyő-file felépítését, hogy az adott karakter legfelső byte-jához tartozó címet töltjük csak be, s majd a gépi kódú rutin az egymás után érkező nyolc karakter-byte és egy attributum-byte megfelelő címét kiszámítja. Ez történik itt a rutinban.
A 40120. címtől ismét felfedezhetjük a ROM egyes részeit (1478. cím), a 40124. címtől kezdődik az 'egy byte felépítése' ciklus, a 40139. címtől pedig maga a LOAD rutin törzse a bitbeolvasó szubrutin (megpiszkálva egy kis keret-"díszítéssel").
A rutin ismeretében fel van adva a lecke: egy olyan programot kell megírnunk, amely egy normál SCREEN$ állományú file-ból a rutinunk számára elfogadható file-t generál.
Ennek a képernyő-file-nak a következőképpen kell a kazettán elhelyezkednie:

karakter legfelső byte képernyőcímének alsó byte-ja /karakter legfelső byte képernyőcímének felső byte-ja /0. adatbyte/ 1. adatbyte /2. adatbyte /3. adatbyte / 4. adatbyte /5. adatbyte /6. adatbyte / 7. adatbyte /attributum byte

(A karakter legfelső byte-ja a 0., a legalsó pedig a 7.)
Ezután következnek egy újabb karakter adatai, és így tovább. Megállapítható, hogy ilyen alapon tetszőleges sorrendben építhetjük fel a képernyőt, erre nehéz lenne egységes útmutatót adni, most csak egyféle verziót ismertetünk, azt, amelynek segítségével egy olyan adatállomány alakítható ki, amivel a képernyőt karaktersoronként fentről folyamatosan lefelé haladva tölthetjük be.
Először is hívjuk be a memóriába az átkódolni kívánt SCREEN$ file-t, próbaképpen bármely olyan játékét, melynek betöltése nem okoz gondot:

```
LOAD""CODE 40000,6912
```

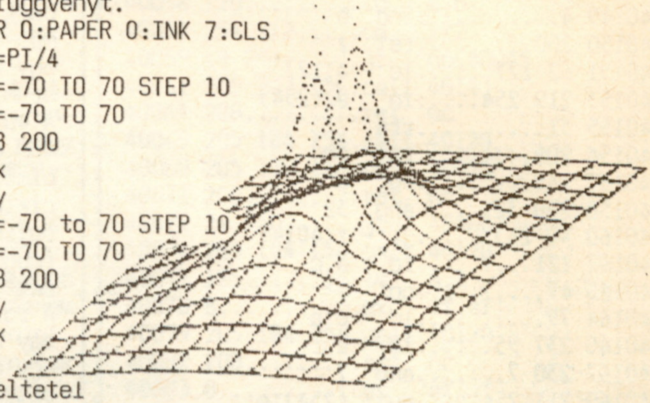
Ezután gépeljük be és futtassuk a 11. oldalon található BASIC programot.

Folytatás a 9.oldalon !

3D térbeli hálók készítése

A matematikával foglalkozók számára érdekes lehet, hogyan állíthatunk elő a képernyőn térbeli görbéket, hálókat, hogyan ábrázolhatunk térbeli függvényeket?
Az itt ismertetett BASIC program a képernyő területét arányosan kihasználva alkalmas a 3D térbeli hálók elkészítésére. A programba a 200. sorba kell elhelyezni a kizáró feltételeket, a 210. sorba pedig a függvényt.

```
10 BORDER 0:PAPER 0:INK 7:CLS
20 LET t=PI/4
30 FOR y=-70 TO 70 STEP 10
40 FOR x=-70 TO 70
50 GO SUB 200
60 NEXT x
70 NEXT y
80 FOR x=-70 to 70 STEP 10
90 FOR y=-70 TO 70
100 GO SUB 200
110 NEXT y
120 NEXT x
130 STOP
200 REM feltetel
210 REM fuggveny
220 PLOT 70+x+(70+y)*COS t,z+(70+y)*SIN t
230 RETURN
```



Példák:

1. Elektropotenciális csúcs

```
200 IF x=0 AND y=0 THEN RETURN

210 LET z=1000/SQR (x*x+y*y)

215 IF z+(70+y)*SIN t>175 THEN RETURN
```

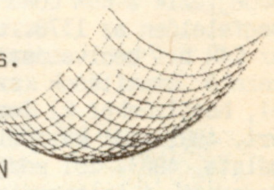
A 215. sor azért kell, hogy a képernyő belső területét elhagyva ne történjen kalkuláció, mivel az hibajelzéshez vezetne!

2. Sarkain felhajtott lemez

A 200. kezdeti feltétel nem szükséges.

```
210 LET z=(x*x+y*y)/150

220 IF z+(70+y)*SIN t>175 THEN RETURN
```

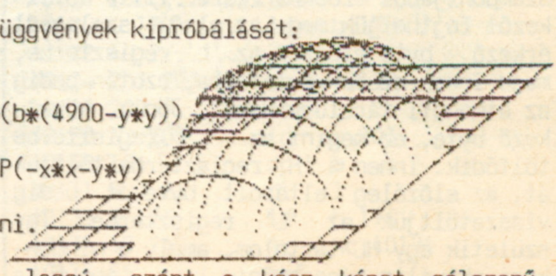


3. Félgömb

```
200 IF(x*x+y*y)>2500 THEN LET z=0: GOTO 220
210 LET z=SQR (2500-x*x-y*y)
220 IF z+(70+y)*SIN t>175 THEN RETURN
```

Ezen túlmenően javasoljuk még a következő függvények kipróbálását:

```
z=a*SIN(x/10)
z=a*(4900-y*y)
z=(a*SIN(x/10))*(b*(4900-y*y))
z=a*x*x*EXP(-x)
z=a*(x*x+y*y)xEXP(-x*x-y*y)
```



ahol "a" tetszőleges érték, lehet próbálkozni.

Megjegyezzük, hogy a program működése igen lassú, ezért a kész képet célszerű SCREEN$ állományként kazettára menteni.

Amstrad vagy Spectrum ?

Sokan feltették ezt a kérdést, amikor megjelent a piacon a 128K Spectrum új verziója, az immár AMSTRAD cég által továbbfejlesztett +2 változat. A Sinclair cég bukását követően az AMSTRAD által forgalmazott gépek szinte egyeduralkodóak voltak Angliában. Idén nyáron Londonban már nagyon sokat kellett járkálnia egy turistának ahhoz, hogy 48K-s gépet tudjon vásárolni, ellenben +2-t minden jobb szaküzletben árultak mindössze 149 Fontért.
Nos, a +2 tetszetős szürke dobozba épített számítógép, kiváló kemény billentyűzettel. A normál 128K-hoz képest alapvető változás, hogy a készülék egy beépített magnetofont is tartalmaz, amit nem akarunk dícsérni, de nem is lehet. A mechanika kemény, nehezen kezelhető, a számláló nagyon hiányzik, és nincs a gépnek 'EAR' csatlakozója sem, így külső magnetofon nem csatlakoztatható. Viszont elhelyezték a gépben az Interface II-t, azaz 2 db Joystick csatlakozót kapott, de - és most figyeljenek! - üzleti taktikai célból csak a géphez hozzá adott, vagy külön vásárolt SJS1 típusú joystick használható, mivel a D9 joystick aljzat bekötését megváltoztatták. No persze ez könnyen kivédhető, de ez esetben a géphez kapott joy-stick-et vagy elajándékozhatjuk a szomszéd kisfiának, vagy tehetünk rá újabb dugót, mert a gyári egybeöntött.
A gép belső felépítése teljesen megváltozott, a portok kiosztása is más lett, a 32K EPROM-ot felváltotta két 16K-s ROM, stb.
Működésében viszont nem sokban tér el elődjétől, így a továbbiakban általános programozástechnikai információkat adunk. Alapvetően azoknak a 128K tulajdonosoknak szánjuk, akik erre már hónapok óta repesve várnak, de úgy gondoljuk, hogy ezek az információk közérdeklődésre is számot tarthatnak.

9.1 HÁROMCSATORNÁS ZENE

Elsősorban a 3 csatornás hang programozását tárgyaljuk részletesen. Ez mind 128K, mind 48K üzemmódban lehetséges, de ez utóbbi esetben csak gépi kódú oldalról (erre a későbbiekben kitérünk). 128K módban BASIC-ből a PLAY utasítás teszi lehetővé a 3 csatornás hang programozását. Kedvcsinálónak gépeljük be az itt közölt mintaprogramot, amely, - ha itt-ott "sántítva" is - lejátsza az 'Ave Maria' kíséretének néhány taktusát.

```
10 LET a$="UX2000W0T10805(3&1gCEgCE)(3&1aDFaDF)(3&1gDFgDF)(3&1gCEgCE)(3&1aEAaEA)(3&1
   #faD#faD)(3&1gDGgDG)(3&1egCegC)(3&1egCegC)(3&1d#fcd#fc)(3&1dgbdgb)(3&1eg#ceg#c)(3
   &1daDdaD)(3&1dfbdfb)(3&1cgCcgC)O4(3&1aCFaCF)(3&1aCFaCF)(3&1gbFgbF)(3&1gCEgCE)(3&1
   $bCE$bCE)(3&1aCEaCE)(3&1aC$EaC$E)(3&1bCDbCD)(3&1gbDgbD)(3&1gCEgCE)(3&1gCFgCF)(3&1
   gbFgbF)(3&1aC#FaC#F)(3&1gCGgCG)(3&1gCFgCF)(3&1gbFgbF)(3&1g$bEg$bE)3&1faCFCaCafafd
   fdO5N3&1gbDFDbDbgbdfedV15N9C"
20 LET b$="O4(1&4_5E)(1&4_5D)(1&4_5D)(1&4_5E)(1&4_5E)(1&4_5D)(1&4_5D)(1&4_5C)(1&4_5C
   )(1&4_5a)(1&4_5b)(1&4_5Sb)(1&4_5a)(1&4_5Sa)(1&4_5g)(1&4_5f)(1&4_5f)(1&4_5d)(1&4_5
   e)(1&4_5g)(1&4_5f)(1&4_5c)(1&4_5f)(1&4_5f)(1&4_5e)(1&4_5d)(1&4_5d)(1&4_5Se)(1&4_5
   e)(1&4_5d)(1&4_5d)(1&4_5c)1&4_5_7c1&4_5_7$c9E"
30 LET c$="O4N7CCCCbbCCCCCCbbbbaaddggggfffeeeeddO3ggCCCCff#f#f$a$aggggggggggggggggc
   c9c9c9c"
40 PLAY a$,b$,c$
```

9.2 A SZABAD MEMÓRIA NAGYSÁGA

Gyakran igen hasznos lehet, ha a 128K üzemmódban meggyőződhetünk a még rendelkezésünkre álló szabad memóriaterület nagyságáról. BASIC-ből a 128K rendszer a 48K-n felüli területeket RAM-disc-ként kezeli, vagyis ide tetszés szerint menthetünk/tölthetünk különböző rutinokat, adatokat, vagy akár több kisebb BASIC programot is. A következő utasítássor segítségével meggyőződhetünk a RAM+RAM-disc szabad memória nagyságáról:

```
PRINT (65535-USR 7962 + (PEEK 23429 + 256 * PEEK 23430 + 65536 * PEEK 23431 ))
```

Alaphelyzetben 118181 byte-ot kapunk eredményül.

10. GÉPI KÓD TANFOLYAM

Ahhoz, hogy egyáltalán gépi kódú programot tudjunk írni, szükségünk van a gépi kódú utasítások ismeretére. Az utasításokat csak kódjukkal igen nehéz lenne értelmezni, ezért minden gépi kódú utasításnak létezik egy közérthetőbb jelkombinációja is. Ezeket mnemonik-oknak nevezzük, és gyakorlatilag az utasítások angol neveinek rövidítései.

Pl. DEC-kód	HEX-kód	jelentés	mnemonik
201	C9	RETURN	RET
0	00	NO OPERATION	NOP
123	7B	LOAD A,E	LD A,E

stb.

Jól látható, hogy a mnemonikok alapján könnyebb a gépi kódú program értelmezése, áttekintése, javítása, stb.
Ezért a programozók olyan programokat is kifejlesztettek amelyek a kódszámsort mnemonikokká fordítják. Ezeket a programokat DISASSEMBLER programoknak hívjuk.

A gépi kódú program értelmezéséhez az a legcélszerűbb ha egymás mellett látjuk az adott memóriahely címét, a hozzátartozó utasítás kódját (kódjait) és az utasítás mnemonik szerkezetét is. például a ROM memória első részének DISASSEMBLÁLT listája hexadecimálisan a következő:

```
0000   F3         DI
0001   AF         XOR A
0002   11 FF FF   LD DE,+FFFF
0005   C3 CB 11   JP +11CB
```

Ugyanez decimálisan:

```
0   243          DI
1   175          XOR A
2   17,255,255   LD DE,65535
5   195,195,17   JP 4547
```

Az itt látható formátumot ASSEMBLY formátum-nak nevezzük.

Az első oszlop a címmező, a második az utasításbyte-ok mezője - vagyis itt annyi számot találunk ahány byte-ból áll az adott utasítás - , és természetesen a címmezőben is ennek megfelelően lép tovább a cím értéke. A harmadik oszlop pedig a mnemonikmező. Maradt még egy oszlop (a negyedik) amiben nem látunk semmit. Ennek a mezőnek COMMENT-(megjegyzés)-mező a neve, és azért nem látunk benne semmit, mert bár programíráskor a forrásnyelvű listában elhelyezhetünk megjegyzéseket - mint pl. a BASIC REM - de ezek a program gépi kóddá fordításakor (ASSEMBLÁLÁS) elvesznek, így DISASSEMBLÁLÁS után a negyedik mező üres.

Mindezek után vágjunk bele a sűrűjébe, hiszen amíg nem ismerjük az utasításokat addig gépi kódú programot sem tudunk írni.

10. GÉPI KÓD TANFOLYAM

A gépi kódú utasítás elvének megértetéséhez BASIC analógiákat fogunk mutatni, de már itt felhívnánk a figyelmet arra, hogy az "egy kalap alatt" közölt BASIC analógia futási eredményét nem ajánlatos összevetni a gépi kódú utasítás futási eredményével. Ez a párosítás ugyanis (már ahol lehetőség nyílik rá) kizárólag szemléltető jellegű és csak azok számára lehet hasznos akik már ismerik a BASIC utasításokat. Természetesen bőven akad olyan gépi kódú utasítás is amelyhez nincs BASIC analógia, még ha netán az eredmény (BASIC-ből) előállítható lenne, nem beszélve az olyan gépi kódú lehetőségekről, amelyeket BASIC-ból egyszerűen lehetetlen megoldani.
Az utasítások decimális és hexadecimális kódjait is közöljük.

I. NOP utasítás

Ez az utasítás egyedülálló!

Kódja a zérus: 00 - NOP

BASIC-ben igen nehéz analógiát találnunk, legjobban a REM hasonlít hozzá, de ez is csak durva összehasonlítás. Ha generálunk egy BASIC programsort és a sorszám mellé csak egy SPACE-t nyomunk, akkor ennek semmi értelme nincs. A program futás közben megnézi, végrehajtható utasítást nem talál, de hibaüzenettel sem áll le. Nos a NOP-nál is hasonló a helyzet. Amikor az utasításregiszterbe zérus kerül, a processzor az utasítást úgy hajtja végre, hogy a programszámlálót tovább lépteti és bekéri a következő adatbyte-ot.

Leginkább arra használhatjuk, hogy gépi kódú programunkban helyet biztosítsunk később elhelyezendő adat(ok) számára. Erre még visszatérünk a programozási stratégiák tárgyalásakor.

II. Töltő utasítások

Ezek az utasítások a gépi kódú programok alapkövei. Egyébként adatok töltését oldják meg memóriából-regiszterbe, regiszterből-regiszterbe ill. regiszterből memóriába. Mivel igen sok töltő utasítás létezik, különböző alcsoportokat vezetünk be.

1. Regiszter töltése konstans adatbyte-tal

A konstans értéke (mint tudjuk) 0-255-ig terjedhet.

Ebbe a csoportba 7 utasítás tartozik, ezek a következők:

62,n	3E N	LD A,N	; konstans töltése az A regiszterbe
38,n	26 N	LD H,N	; konstans töltése a H regiszterbe
46,n	2E N	LD L,N	; konstans töltése az L regiszterbe
6,n	06 N	LD B,N	; konstans töltése a B regiszterbe
14,n	0E N	LD C,N	; konstans töltése a C regiszterbe
22,n	16 N	LD D,N	; konstans töltése a D regiszterbe
30,n	1E N	LD E,N	; konstans töltése az E regiszterbe

Az első oszlopban decimálisan, a másodikban hexadecimálisan láthatjuk az utasítás-byte-ok kódjait. A harmadik oszlopban a mnemonikokat találjuk, és íme a negyedikben a megjegyzés.
n a decimális, N pedig a hexadecimális adatbyte jelzése (a mnemonik oszlopban most tetszőleges).

Ha pl. az E regiszterbe akarunk 163-at tölteni, úgy az utasítás formája:

```
30,163    1E A3    LD E,163  ; 163 töltése az E regiszterbe
```

BASIC-ben az értékadáshoz hasonlítható a legjobban:

```
LET E=163
```

vagyis a végrehajtás után E felveszi a 163-as értéket, csak gépi kódban E egy regiszter neve és nem egy változó, s maximális értéke is csak 255 lehet míg BASIC-ben ennél nagyobb is.

Ezek az utasítások a jelzőbiteket nem változtatják meg, ezért itt ezzel a témával még nem foglalkozunk.

2. Regiszterpár töltése konstanssal

Ebbe a csoportba 6 utasítás tartozik, ezek a következők:

```
    33,n,n     21 N N     LD HL,NN  ; kétbyte-os adat töltése a
     1,n,n     01 N N     LD BC,NN  ; kiválasztott regiszterpárba
    17,n,n     11 N N     LD DE,NN
221,33,n,n     DD 21 N N  LD IX,NN
253,33,n,n     FD 21 N N  LD IY,NN
    49,n,n     31 N N     LD SP,NN
```

Ha pl. DE regiszterbe 16384-et (a képernyő kezdőcímét) akarjuk tölteni, akkor az utasítás formája a következő:

```
17,0,64    11 00 40    LD DE,16384
```

Na itt álljunk meg egy pillanatra. Az előzőekben már volt szó a hexadecimális cím és adatkezelés könnyebb voltáról, ezért is nem közöljük külön oszlopban a hexadecimális mnemonik mezőt. Lényeg a lényeg: kétbyte-os cím v. adat esetén (hexadecimális számrendszerben) az adatbyte mezőben lévő második byte kerül előre, az első pedig mögé, vagyis ebben az esetben ez így festene:

```
LD DE,4000
```

Mindenképpen itt kell megemlítenünk az alsó és felső byte jelentőségét is. Az adatbyte mezőben a kétbyte-os adatnak mindig az alsó byte-ja van elöl, a felső pedig mögötte. Mindig az alsó byte töltődik a regiszterpár második betűjével jelölt regiszterébe (jelenleg tehát zérus az E regiszterbe), és a felső byte töltődik a regiszterpár első betűjével jelölt regiszterébe (jelenleg tehát 64 (HEX 40) a D regiszterbe.

Decimálisan az érték úgy számítható ki, hogy a felső byte értékét meg kell szoroznunk 256-tal és hozzá kell adni az alsó byte értékét: azaz

```
64*256+0=16384
```

Szemléltető példánk BASIC-ben:

```
LET E=0
LET D=64
LET DE=256*D+E
```

Az utasítások a jelzőbiteket nem állítják.

3. Regisztertartalmak cseréje ill. átmásolása

Ebbe a csoportba 59 utasítás tartozik, ebből 49 szorosan összetartozó így ezekről érdemes összefoglaló táblázatot készíteni. A többi 10 kilóg a sorból és külön tárgyaljuk.

	x=A	x=H	x=L	x=B	x=C	x=D	x=E
LD A,x	127/7F	124/7C	125/7D	120/78	121/79	122/7A	123/7B
LD H,x	103/67	100/64	101/65	96/60	97/61	98/62	99/63
LD L,x	111/6F	108/6C	109/6D	104/68	105/69	106/6A	107/6B
LD B,x	71/47	69/44	70/45	65/40	66/41	67/42	68/43
LD C,x	79/4F	76/4C	77/4D	72/48	73/49	74/4A	75/4B
LD D,x	87/57	84/54	85/55	80/50	81/51	82/52	83/53
LD E,x	95/5F	92/5C	93/5D	88/58	89/59	90/5A	91/5B

Ezek az utasítások a jelzőbitekre nincsenek hatással.

Regiszterek közötti töltés lehetséges még:

- Az I és az R regiszterekkel kapcsolatban:

```
237,87    ED 57    LD A,I    ; I reg. értéke A reg.-be
237,95    ED 5F    LD A,R    ; R reg. értéke A reg.-be
237,71    ED 47    LD I,A    ; A reg. értéke I reg.-be
237,79    ED 4F    LD R,A    ; A reg. értéke R reg.-be
```

Megjegyzés: a P/V jelzőbitet állítják.

- Csere főregiszterpárok között

```
235       EB       EX DE,HL  ; HL értéke DE-be és viszont
```

Érdekes, hogy az utasításkészletben csak ez az egy 'regiszterpár értékeinek cseréje' utasítás létezik.

A jelzőbiteket nem befolyásolja.

- Csere fő- és segédregiszterek között:

```
217       D9       EXX
  8       08       EX AF,AF'
```

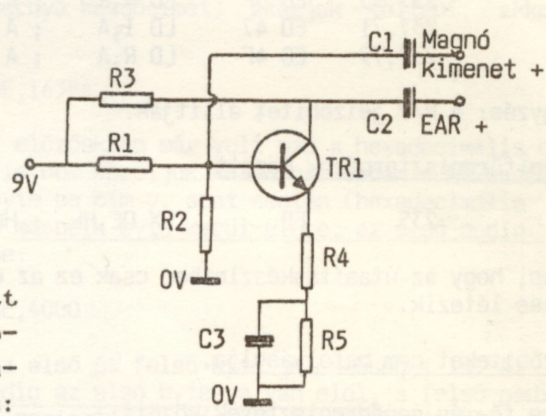
Az első komplett megcseréli a fő és segédregiszterek tartalmát, míg a második ezt csak az AF és AF' regiszterekkel teszi meg. Ez utóbbit elsősorban a jelzőbitek tárolására szokás alkalmazni.
Az említett utasítások BASIC analógiája a változók közötti manipuláció:

```
LET B=20
LET E=10
LET CS=B:LET B=E:LET E=CS
```

Mint látjuk a cseréhez BASIC-ben egy újabb - átmeneti - változót kell generálnunk, gépi kódban ezt a processzor automatikusan elvégzi.

TARTALOMJEGYZÉK:





Az előző részben ajánlott ~SAVE gerjedésgátló~ kapcsolási rajzába hiba csúszott. Íme a helyes rajz:

A szerzők a következő címen érhetők el :
SPECTRUM VILÁG - BUDAPEST-3, *Postán maradó*, 1300

Akiknek nem jutott volna a 'SV' első ill. második részéből, azoknak azt ajánljuk, rendeljék meg a fent olvasható címen!

Felelős kiadó: Rucz Lajos, Székely László K. F. eng. sz. 59406
10000 – 708152 Pátria Nyomda
Felelős vezető: Vass Sándor vezérigazgató

Ára: 39,- Ft



ALI PJT
TÖRÖK KISVENDÉGLŐ
BUDAPEST XII.,
MÁRVÁNY U. 24/B.
12h-21h-ig
TÖRÖK
OLASZ
MAGYAR
ÉTELEK
SZÜNNAP: VASÁRNAP